



(19) RU⁽¹¹⁾ 2 189 072⁽¹³⁾ C2

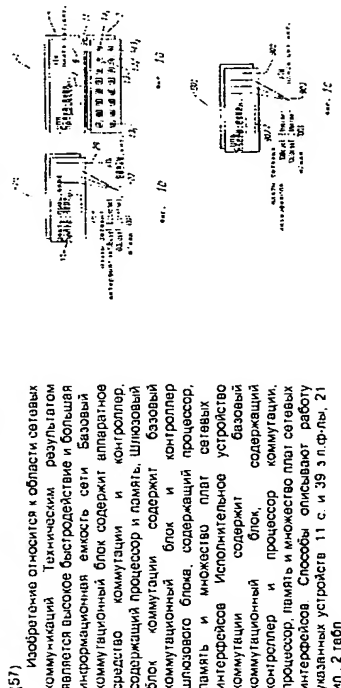
(51) МКП⁷ G 06 F 13/14, H 04 L 12/56,

12/66

РОССИЙСКОЕ АГЕНТСТВО
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ

- (21) Заявка, 98116258/09, 30.01.1997
(24) Дата начала действия патента, 30.01.1997
(30) Приоритет 31.01.1996 US 08/597,520
22.11.1996 US 60/024,272
(43) Дата публикации заявки, 20.06.2000
(46) Дата публикации, 10.09.2002
(56) Ссылки: US 5379297 A, 03.01.1995, RU 2035065
CI, 10.05.1995, US 5444702 A, 22.08.1995, EP
0472066 A1, 04.03.1992, EP 051144 A1,
28.10.1992, SU 1695329 A1, 30.11.1991.
(85) Дата первого заявления РСТ на национальную
фазу: 31.08.1998
(86) Заявка РСТ:
US 97/01595 (30.01.1997)
WO 97/28505 (07.08.1997)
(87) Публикация РСТ
WO 97/28505 (07.08.1997)
(98) Адрес для переписки:
129010, Москва, ул. Б. Спасская, 25, стр.3,
ООО "Юридическая фирма Городецкая и
Партнеры", Ю.Д.Кузнецову, рег. № 595



RU 2 189 072 C 2



(19) RU⁽¹¹⁾ 2 189 072⁽¹³⁾ C2

(51) МКП⁷ G 06 F 13/14, H 04 L 12/56,

12/66

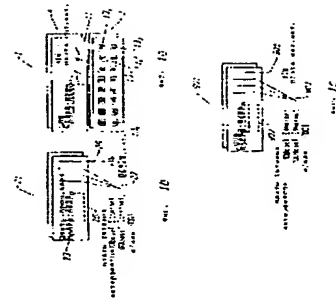
RUSSIAN AGENCY
FOR PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

- (21) (22) Application: 98116258/09, 30.01.1997
(24) Effective date for property rights: 30.01.1997
(30) Priority: 31.01.1996 US 08/597,520
22.11.1996 US 60/024,272
(43) Application published: 20.06.2000
(46) Date of publication: 10.09.2002
(85) Commencement of national phase: 31.08.1998
(86) PCT application:
US 97/01595 (30.01.1997)
(87) PCT publication:
WO 97/28505 (07.08.1997)
(98) Mail address:
129010, Moskva, ul. B. Spasskaja, 25, str.3,
ООО "Юридическая фирма Городецкая и
Партнеры", Ю.Д.Кузнецову, рег. № 595

(54) IMPROVED METHOD AND DEVICE FOR DYNAMIC SHIFT BETWEEN ROUTING AND SWITCHING BURSTS
IN DATA TRANSMISSION NETWORK

(57) Abstract:
FIELD: network communications.
SUBSTANCE: base switching unit has hardware switching facility and controller; the latter has processor and memory. Gateway switching unit has base switching unit and gateway unit controller that has processor, memory, and plurality of network interface boards. Actuating switching device has base switching unit incorporating switching controller and processor, memory, and plurality of network interface boards. Method describes operation of these devices. EFFECT: enhanced speed and information capacity of network. 50 cl, 21 dwg, 2 tbl



RU 2 189 072 C 2

Коммуляторы локальных сетей предоставляют относительно экономичный и высокоэффективный способ решения проблемы перегрузки в сегментах локальных сетей, совместно использующих носители информации. Технология коммутации представляется более эффективными средством в управлении трафиком и распределении ширины полосы в пределах локальной сети, чем концентраторы.

Режим асинхронной передачи (АТМ) характеризует собой высокоскоростное, мультиплексированное, многоадресное, текстовое, реляционное, в настоящее время в сетях с использованием каналов современных систем, не использующих маршрутизаторы. АТМ представляет собой высокоразвитую технологию плавной передачи, настолько отличную от современных сетевых архитектур, таких как IP, что в настоящее время отсутствуют простой путь внедрения данной технологии. АТМ испытывает затруднения в эволюционной поддержке существующего локального сетевого трафика ввиду его ориентированности на соединения архитектуры, что создает необходимость в дополнительной разработке очень сложных протоколов. Такие протоколы создадут очевидные проблемы неприятием большого времени установления соединения коммутируемого виртуального канала. Кроме того, обеспечение возможности использовать TCP/IP подразумевает и признать АТМ средством передачи.

✦

В соответствии с одним из вариантов осуществления настоящего изобретения способ передачи пакетов между узлом восходящего потока и узлом нисходящего потока заключается в следующем. Узел нисходящего потока находится в нисходящем состоянии относительно узла восходящего потока. Способ предусматривает этап установления по упомянутым виртуальным каналам между узлом восходящего потока и узлом нисходящего потока, приема пакета в узле нисходящего потока и выполнения операции классификации пакета в узле нисходящего потока по пакету, чтобы определить, принадлежит ли пакет к специфическому каналу, который должен быть перенаправлен в узел восходящего потока. Способ также предусматривает этап

В соответствии с еще одним преимуществом осуществления настоящей изобретения обеспечивается коммутируемый межсетевой блок (шлюзовый блок) в системе для передачи пакетов в сети. Система содержит базовый коммутируемый блок, связанный с коммутируемым шлюзовым блоком через канал связи. Коммутируемый шлюзовый блок включает в себя контроллер шлюза и программное обеспечение, которое позволяет шлюзовому блоку проследить, память и множество плат сетевых интерфейсов (NIC). Контроллер шлюза обеспечивает на материальном носителе возможность коммутирующему шлюзовому блоку передавать поток пакетов к базовому коммутируемому блоку, чтобы обеспечить динамическое смещение между маршрутизацией и коммутируемой логикой для оптимизации пропускной способности трафика.

осуществления настоящего изобретение обеспечивается средство коммутации в системе для передачи пакетов в сети. Система содержит базовый коммутационный блок, связанный со средством коммутации через канал связи, причем базовый коммутационный блок содержит контроллер и коммутатор. Средство коммутации включает в себя процессор, память и множество плат сетевых интерфейсов (NIC), причем каждая плата из этих плат NIC обеспечивает канал связи и, по меньшей мере, одна из этих плат NIC имеет возможность соединения, по меньшей мере, с одним узлом сети. Коммутационное средство также содержит машиночитаемый программный код на материальном машиночитаемом носителе в памяти, позволяющий контроллеру базового коммутационного блока идентифицировать поток и перенаправлять этот поток пакетов из первого узла во второй узел в сети, а также позволяет контроллеру базового коммутационного блока выдавать команды средству коммутации осуществлять пересылку пакета данного потока из первого узла во второй узел через коммутатор. Соответственно функция пересылки пакетов не затрудняет контроллер базового коммутационного блока.

Указанные и другие варианты осуществления изобретения вместе с их признаками и преимуществами описаны подробно в нижеприведенном тексте, иллюстрирующем чертежи.

Краткое описание чертежей

Фиг. 1а - упрощенная диаграмма базового коммутационного блока, соответствующей возможному варианту осуществления изобретения; на фиг. 1б показана упрощенная диаграмма коммутационного шлюзового блока системы, соответствующей другому варианту осуществления изобретения;

Фиг. 1с - упрощенная диаграмма средства коммутации в системе, соответствующей еще одному варианту осуществления изобретения;

Фиг. 2а-2с - упрощенные диаграммы примерных сетей, конфигураций в соответствии с вариантами осуществления изобретения;

Фиг. 3 - обобщенная блок-схема компьютерной системы, используемой в соответствии с вариантами осуществления изобретения;

Фиг. 4 - обобщенная блок-схема АТМ коммутатора, используемого в соответствии с вариантами осуществления изобретения;

Фиг. 5а - упрощенная диаграмма, иллюстрирующая в обобщенном виде процедуру инициализации в каждом системном узле, согласно возможному варианту осуществления изобретения;

Фиг. 5б - упрощенная диаграмма, иллюстрирующая работу системного узла;

Фиг. 5с - упрощенная процедура, иллюстрирующая в обобщенном виде процедуру в средстве коммутации, когда пакет поступает в один из интерфейсов после инициализации;

Фиг. 5д - упрощенная диаграмма, иллюстрирующая процедуру в контроллере

посредством линии связи, по меньшей мере, одно средство коммутации, например, с использованием коммутатора контроллера (коммутации), когда пакет поступает от средства коммутации на один из его интерфейсов по каналу, установленному по умолчанию, после инициализации;

Фиг. 6а - диаграмма, иллюстрирующая в обобщенном виде этапы, используемые для маршрутизации потока в системном узле;

Фиг. 6б - диаграмма, иллюстрирующая в обобщенном виде этапы, используемые для коммутации потока в базовом коммутационном узле;

Фиг. 6с - диаграмма, иллюстрирующая в обобщенном виде этапы, используемые для пересылки пакета в системном узле (ити в узле коммутации);

Фиг. 6д - диаграмма, иллюстрирующая в обобщенном виде этапы, выполняемые в контроллере коммутации, при маршрутировании потока для пакетов, принимаемых от средства коммутации источника, в трех ситуациях;

Фиг. 6е - диаграмма, иллюстрирующая в обобщенном виде этапы, выполняемые в контроллере коммутации, при маршрутировании потока для пакетов, принимаемых от назначенного узла для коммутации и предназначенных для интерфейса в назначенном средстве коммутации;

Фиг. 7а-7б - форматы идентификаторов потока для потока типа 1 и для потока типа 2;

Фиг. 8а - структура сообщения обобщенного протокола близости IFMP (протокол управления потоком Ispon), согласно одному из вариантов осуществления изобретения;

Фиг. 8б - обобщенный пакет протокола IP (а его современная версия IPv4) с полем данных переменной длины, в который может быть инкапсулировано сообщение IFMP;

Фиг. 8с - упрощенная диаграмма, иллюстрирующая работу системного узла после приема пакета с входящим сообщением протокола близости IFMP;

Фиг. 8д - упрощенная диаграмма, иллюстрирующая работу передающего системного узла, если входящее сообщение протокола близости IFMP не является протоколом RSTACK;

Фиг. 9а - структура обобщенного сообщения протокола перенаправления IFMP в соответствии с одним из вариантов осуществления изобретения;

Фиг. 9б - обобщенная диаграмма, описывающая работу системного узла после приема сообщения протокола перенаправления IFMP;

Фиг. 9с-9д - структуры элемента сообщения REDIRECT (перенаправление) элемента сообщения RECLAIM (восстановление) элемента сообщения RECLAIM /ACK (подтверждение восстановления), элемента сообщения LABEL RANGE (интервал метки), элемента сообщения ERROR (ошибка) в теге сообщения 394 соответствующих сообщений протокола перенаправления IFMP;

Фиг. 10а - формат поля метки в линии передачи АТМ данных, согласно одному из вариантов осуществления изобретения;

Фиг. 10б -10с - иллюстрации соответственно пакетов, инкапсулированных

потока типа 0, потока типа 1 и потока типа 2, согласно одному из вариантов осуществления настоящего изобретения;

Фиг. 11а - формат инкапсулированного пакета протокола IFMP-C и сообщении запроса "удалить переход" IFMP-C по протоколу управления коммутацией;

Фиг. 11б - формат сообщения протокола близости GSMР;

Фиг. 11с - упрощенная диаграмма, иллюстрирующая работу передающего узла после приема пакета с входящим сообщением протокола близости GSMР;

Фиг. 11д - диаграмма состояния, иллюстрирующая работу передающего узла, если входящее сообщение протокола близости GSMР не является сообщением RSTACK;

Фиг. 12 - формат обобщенного сообщения управления соединением протокола GSMР;

Фиг. 13а -13е - упрощенные диаграммы, иллюстрирующие работу приемного узла после приема сообщений "добавить переход", "исключить переход", "исключить дерево", "проверить переход" и "удалить все" управления соединениями протокола GSMР соответственно;

Фиг. 13г - формат сообщения "переместить корень" управления соединениями протокола GSMР;

Фиг. 13и - формат сообщения "переместить переход" управления соединениями протокола GSMР;

Фиг. 13л - упрощенная диаграмма, иллюстрирующая работу передающего узла после приема пакета с входящим сообщением "переместить корень" управления соединениями протокола GSMР;

Фиг. 13н - формат сообщения "переместить переход" управления соединениями протокола GSMР;

Фиг. 13п - упрощенная диаграмма, иллюстрирующая работу передающего узла после приема пакета с входящим сообщением "переместить переход" управления соединениями протокола GSMР;

Фиг. 14 - формат сообщения управления портом протокола GSMР;

Фиг. 15а - иллюстрация инкапсулированного пакета 1000 протокола IFMP-C;

Фиг. 15б - общая структура типового сообщения IFMP-C 1002, которое может содержаться в поле 1006 сообщения инкапсулированного пакета 1000 IFMP-C по фиг. 15а;

Фиг. 16а - структура сообщения 1040 протокола близости IFMP-C, которое может содержаться в поле 1006 сообщения IFMP-C инкапсулированного пакета 1000 IFMP-C по фиг. 15а;

Фиг. 16б - диаграмма состояния, иллюстрирующая работу передающего узла (контроллера IFMP-C или исполнительного устройства (последствия) IFMP в трех различных состояниях протокола близости IFMP-C;

Фиг. 17а и 17б - структуры сообщений запроса и ответа перечня интерфейса IFMP-C соответственно;

Фиг. 7с и 17д - структуры сообщений запроса и ответа очереди интерфейса IFMP-C соответственно;

Фиг. 17е - структура сообщения 1170 запроса конфигурации интерфейса IFMP-C;

Фиг. 18а - формат 1200 сообщений запроса "добавить переход" протокола IFMP-C и сообщений запроса "удалить

Фиг. 18б - поле 1240 преобразования данных для типа преобразования "учебная пакета" в сообщении запроса "добавить переход" протокола IFMP-C и сообщении запроса "удалить переход" IFMP-C по фиг. 18а;

Фиг. 18 с - формат 1250 сообщений ответа "добавить переход" протокола IFMP-C и сообщении ответа "удалить переход" IFMP-C;

Фиг. 18д - структура сообщения 1260 запроса "удалить дерево" протокола IFMP-C;

Фиг. 18е - структура сообщения 1300 запроса "переместить переход" протокола IFMP-C;

Фиг. 19а - структура сообщения 1400 запроса "получить статистику дерева" протокола IFMP-C;

Фиг. 19б - структура 1405 полей "данные дерева", иллюстрирующая используемые поля "данных дерева";

Фиг. 20а и 20б - структура сообщения 1420 запроса "считать переход" протокола IFMP-C и сообщения 1430 ответа "считать переход" протокола IFMP-C соответственно;

Фиг. 21а - структура сообщения 1440 запроса "информация узла" протокола IFMP-C;

Фиг. 21б и 21с - структура сообщения 1460 запроса "статистика интерфейса" протокола IFMP-C и сообщения 1470 ответа "статистика интерфейса" протокола IFMP-C соответственно;

Фиг. 21д - структура поля 1480 "статистика интерфейса" в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21е - структура поля 1494 "общая статистика" в поле 1480 "статистика интерфейса" в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21г - структура поля 1530 "специальная статистика" (для АТМ интерфейса) в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21и - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21л - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21м - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21н - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21о - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21п - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21к - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21л - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21м - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21н - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21о - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

Фиг. 21п - структура поля 1540 "специальная статистика" (для интерфейса сети "Интернет") в поле "статистика интерфейса" 1480 в сообщении 1470 ответа "статистика интерфейса" протокола IFMP-C по фиг. 21с;

5 10 15 20 25 30 35 40 45 50 55 60

как уровень передачи данных (уровень 2 модели OSI (протокол межсетевых открытых систем)).

Машину 3 коммутации служат для выполнения функций высокоскоростной коммутации, если это требуется базовым коммутационным блоком, как это определяется системным программным обеспечением 7. Функциональные возможности коммутационных систем коммутации ограничены только аппаратными средствами, используемыми в составе машины 3 коммутации. Соответственно рассматриваемый вариант осуществления изобретения обеспечивает получение таких преимуществ АТМ технологии, как высокая производительность, высокая пропускная способность и широкая полоса. Разумеется, другие технологии коммутации, такие как высокоскоростная пакетная коммутация, Ethernet и другие, также могут быть использованы в зависимости от конкретного применения.

В одном из вариантов осуществления настоящего изобретения контроллер 5 коммутации представляет собой компьютер, соединенный с аппаратными средствами 3 АТМ коммутатора через канал АТМ, в котором программное обеспечение системы установлено в компьютере. Кроме выполнения стандартных функций маршрутизации согласно протоколу IP, на требуемое соединение, контроллер 5 коммутации также принимает решения о классификации пакетов на локальной основе.

Как показано на фиг.1b, шлюзовый блок коммутации 21 системы коммутации, соответствующий другому варианту осуществления настоящего изобретения, содержит контроллер 23 коммутации шлюза, системное программное обеспечение 25, установленное в контроллере 23 коммутации шлюза. Контроллер 23 коммутации шлюза содержит множество сетевых адаптеров или плат NIC 27 и АТМ плату NIC 29. Аналогично контроллеру 5 коммутации базового коммутационного блока 1 контроллер 23 коммутации шлюза также представляет собой компьютер, оснащенный АТМ платой NIC 29, имеющий системное программное обеспечение 25, установленное в компьютере. Как описано выше, шлюзовый блок коммутации 21 служит в качестве устройства доступа для обеспечения соединения существующей локальной сети и основной среды с сетью базовых коммутационных блоков. Соответственно платы NIC 27 могут быть различных типов, например Ethernet NICs, 100 BaseT Ethernet NICs, FDDI NICs и другие, а также использование конкретных плат NICs 27 зависит от типов существующих локальных сетей и основной операционной среды, в которой обеспечивается доступ посредством шлюзового блока коммутации 21. Ясно, что множество локальных сетей могут быть соединены со шлюзовым блоком коммутации 21. АТМ плата NIC 29 позволяет шлюзовому блоку коммутации 21 соединяться через АТМ канал с базовым коммутационным блоком 1. Разумеется, плата NIC 27 может представлять собой АТМ плату NIC для

5 10 15 20 25 30 35 40 45 50 55 60

коммутационного блока. Соответственно в некоторых ситуациях, когда средства независимого управления направлением потока не требуются в определенном элементе или если функциональные возможности базового коммутационного блока могут быть использованы более эффективно, указанный последний оказывается пригодным для использования. Такой порядок может также использоваться в качестве экономичной замены шлюзового блока коммутации. Система совместима с протоколом Интернет (IP) в его текущей версии (IPv4), а также с перспективными версиями (например, IPv6). Система обеспечивает динамическое смещение между коммутацией и маршрутизацией пакетов в сети для обеспечения оптимального высокоскоростного распределения пакетов, позволяя избежать при этом "узких мест".

Как показано на фиг.1a, базовый коммутационный блок 1 системы коммутации, согласно возможному варианту осуществления настоящего изобретения, содержит машину 3 коммутации, контроллер 5 коммутации, программное обеспечение 7 системы, установленное в контроллере 5 коммутации. В частности, машина 3 коммутации использует обычные аппаратные средства коммутации асинхронного режима передачи (АТМ). Разумеется, для реализации машины 3 коммутации в настоящем изобретении могут быть использованы и другие технологии, например высокоскоростная пакетная коммутация, Ethernet и другие, в зависимости от конкретного применения. В рассмотряемом варианте осуществления машина 3 коммутатор представляет собой АТМ коммутатор. Любое программное обеспечение, обычно связанное с АТМ коммутатором, соответствующее уровню АТМ адаптации типа 5 (AAL-5), полностью исключено. Таким образом, сигнализация, любой известный протокол маршрутизации, любой сервер эмуляции локальной сети или серверы определения адресов и т.п. исключены. Контроллер 5 коммутации представляет собой компьютер, имеющий адаптер АТМ сети или плату сетевого интерфейса (NIC) 9, соединенную с машиной 3 коммутации посредством АТМ канала связи 11. Программное обеспечение 7 системы установлено в базовом коммутационном блоке 1, более конкретно в компьютере, служащем в качестве контроллера 5 коммутации.

Машину 3 коммутации базового коммутационного блока 1 имеет множество физических портов 13, обеспечивающих соединение с различными устройствами, включая аппаратуру терминалов данных (DTE), аппаратуру передачи данных (DCE), серверы, коммутаторы, шлюзы и т.п. Каждый из физических портов 13, может быть связан посредством АТМ канала с устройством, оснащенным АТМ адаптером или NIC, или с портом другого базового коммутационного блока, или с портом шлюзового блока коммутации, или с портом исполнительного механизма (последним). Аппаратные средства АТМ коммутации, предусматривающие машину 3 коммутации

5 10 15 20 25 30 35 40 45 50 55 60

сигналов, видеосигналов, сигналов данных в локальной сети (LAN), в сетях метрополит (MAN), в расширенных сетях (WAN), Интернет и т.п. Однако изобретение не ограничивается указанными типами сетей. Изобретение может использоваться в самых разнообразных системах, где требуется передача пакетов по сети.

Система, описанная ниже, представляет собой систему динамической коммутации и маршрутизации. Эта система описывается в принципе как "система коммутации", однако следует иметь в виду, что данная система динамически обеспечивает как коммутацию на уровне 2 канала передачи данных, так и маршрутизацию и пересылку пакетов на сетевом уровне 3. Кроме того, базовый коммутационный блок" системы также динамически обеспечивает как коммутацию на уровне 2, так и маршрутизацию и пересылку пакетов на уровне 3. Механизмы (шлюзовый) блок коммутации системы служит в качестве устройства доступа, обеспечивая соединение существующей локальной сети и основной среды с сетью базовых коммутационных блоков. Аналогично шлюзовому блоку коммутации "исполнительное устройство (посредник) коммутации" также служит устройством доступа, предназначенным для обеспечения соединения существующей локальной сети и основной среды по меньшей мере с одним базовым коммутационным блоком. Как шлюзовый блок коммутации, так и базовый коммутационный блок имеют независимые средства управления перенаправлением пакетов, протоколы маршрутизации и принимают решения о маршрутизации независимо от отсутствия команд перенаправления потока, как описано ниже. Шлюзовый блок коммутации и базовый коммутационный блок являются поэтому равноправными. В противоположность этому исполнительные устройства коммутации, не имея независимых средств управления направлением потока, пересылают пакеты, основываясь на командах от базового коммутационного блока, действующего в качестве задющего блока по отношению к упомянутому посреднику. Работа в соответствии с командами от базового коммутационного блока, исполнительное устройство коммутации может пересылать пакеты, принятые от базового коммутационного блока, так что большая часть пакетов, направляемых базовым коммутационным блоком, может в данном случае направляться посредством в существующую локальную сеть или в основную операционную среду по интерфейсам посредника. Такие среды могут включать сети Ethernet, FastEthernet, FDDI (локальная сеть оптического стандарта), Gigabit Ethernet и другие типы локальных сетей. Поскольку так пересылаются пакеты выполняется посредством на базе команд пересылки пакетов, базовый коммутационный блок может иметь больше времени на выполнение других задач, связанных с выполнением протоколов маршрутизации, а также снизить время ожидания при пересылке пакетов. Выполнение задач пересылки пакетов посредством сокращает нагрузку

фиг. 2в-2с показаны конфигурации ряда сетей, которые могут быть реализованы в соответствии с настоящим изобретением. Разумеется, фиг. 2в-2с иллюстрируют лишь приведенные для примера конфигурации, причем возможны много другие примеры конфигураций.

На фиг. 2а показана упрощенная диаграмма конфигурации локальной сети для рабочей (территориальной) группы, в которой базовый коммутационный блок 1 служит в качестве устройства централизованной пересылки пакетов по протоколу IP для всей локальной сети с несколькими шлюзовыми блоками 21 коммутации, обеспечивающими подключение к существующим локальным сетям. Базовый коммутационный блок 1 соединен с группой серверов, включающей три сервера 31_n (где n = 1...3). Каждый сервер 31_n, оснащенный комплектом системного программного обеспечения и ATM платой NIC для обеспечения соединения с базовым коммутационным блоком 1 через соответствующие ATM каналы 33_n, (где n=1...3), представляющие собой каналы OC-3 (155 Мб/с), с использованием серверов, непосредственно связанных с базовым коммутационным блоком 1 через высокоскоростные ATM каналы, обеспечивается ускорение пропускания пакетов для серверов с наиболее частым доступом. Базовый коммутационный блок 1 также соединен с тремя шлюзовыми блоками 21 коммутации через соответствующие ATM каналы 33_n (где n=1...6), также представляющие собой каналы OC-3. Первый шлюзовый блок 21 коммутации, соединенный с базовым коммутационным блоком 1 через канал 33₄, также соединен с основной операционной средой 35, локальной сети, которая может представлять собой некоторый тип сети Ethernet или FDDI, через соответствующий канал 39₁. Основная операционная среда 35, локальной сети связана с персональными компьютерами, терминалами, рабочими станциями 41 через соответствующие платы NICs 43. Аналогичным образом второй и третий шлюзовые блоки коммутации 21, соединенные с базовым коммутационным блоком 1 через каналы 33₅, 33₆, также соединены с основными операционными средами 35₂, 35₃ локальных сетей через соответствующие каналы сети Ethernet или FDDI 39₂, 39₃. Конфигурация по фиг. 2а таким образом позволяет пользователям непосредственно соединяться с различными локальными сетями для периодов без задержки потока трафика протокола IP, не создавая перегрузок, в соответствии с настоящим изобретением.

В качестве другого примера на фиг. 2б показана упрощенная диаграмма конфигурации рабочей группы. Фиг. 2б иллюстрирует высокоскоростную среду рабочей группы, в которой несколько главных компьютеров ("хостов") 45 соединены посредством ATM каналов 33_n, с множеством базовых коммутационных блоков 1, которые соединены со шлюзовым блоком 21 коммутации, соединенным с локальной сетью 35, в которую включены пользователи устройств 41. В этой конфигурации первый

вторым базовым коммутационным блоком 1 посредством ATM канала 33₁ (155 Мб/с). Множество главных компьютеров 45 соединены с первым базовым коммутационным блоком 1 посредством соответствующего ATM канала 33_n, со скоростью 155 Мб/с (где n = 2...5) через соответствующую ATM плату NIC 47. Кроме того, множество главных компьютеров 45 соединены со вторым базовым коммутационным блоком 1 посредством соответствующего ATM канала 33_n, со скоростью 25 Мб/с (где n=8...10) через соответствующую ATM плату NIC 49. Как описано выше, главные компьютеры 45 оснащены ATM платами NICs, с установленными в них комплектами системного программного обеспечения, что позволяет главным компьютерам, использующим протоколы TCP/IP, непосредственно соединяться с базовым коммутационным блоком. Первый и второй базовые коммутационные блоки 1 соединены с шлюзовым блоком 21 коммутации посредством ATM канала 33₆ (155 Мб/с) и 33₇ (25 Мб/с) соответственно.

Соединение первого и второго базовых коммутационных блоков 1 с шлюзовыми блоками 21 коммутации через канал 39 Ethernet или FDDI обеспечивает возможность пользователям главных компьютеров 45 осуществлять информационный обмен с пользователями устройств 41, соединенными с локальной сетью 35. Пользователи устройств 41 могут представлять собой персональные компьютеры, терминалы, рабочие станции, имеющие соответствующие платы NIC 43 для соединения с локальной сетью 35 типа Ethernet или FDDI. Рабочая группа главных компьютеров тем самым оказывается интегрированной без каких-либо препятствий в остальную часть территориальной (например, университетской) сети.

На фиг. 2с представлена конфигурация, использующая базовый коммутационный блок 1, несколько исполнительных устройств 91₁, 91₃, 91₅ коммутации и системный узел 91₆ (например, другой базовый коммутационный блок 1, шлюзовый блок коммутации или главный компьютер). Разумеется, в других конфигурациях могут использоваться дополнительные системные узлы и иные комбинации. На фиг. 2с показано несколько исполнительных устройств 91₁, 91₃, 91₅ коммутации, каждое из которых имеет соответствующие интерфейсы с различными локальными сетями Ethernet 91₇ n (где n=1...8 в данном конкретном примере), каждое имеет последовательные пользовательские устройства (не показаны) и каждое из исполнительных устройств соединено посредством ATM каналов 91₉_n (где n=1...3 в данном конкретном примере) с базовым коммутационным блоком 1, который включает в себя контроллер 921 коммутации, соединенный ATM каналом 923 с машинной коммутацией 925. Разумеется, локальные сети 91₇_n могут представлять собой 10BaseT или 100 BaseT Ethernet, FDDI, Gigabit Ethernet, другие типы сетей или комбинации этих типов сетей. В качестве пользовательских устройств, соединенных с локальной сетью

компьютеры, терминалы, принтеры, серверы, рабочие станции и т.п., имеющие соответствующие платы NICs для соединения с локальной сетью 91₇_n. Системный узел 91₆ базового коммутационного блока 1 соединен с машинной коммутацией 925 посредством ATM канала связи 91₉.

В общем случае контроллер коммутации 921 (фиг. 2с) осуществляет управление исполнительными устройствами коммутации путем управления их интерфейсами (для передачи и приема пакетов) и путем приема/передачи информации о состоянии пакетов, принимаемых в конкретный поток из специфических типов потоков. Конкретные типы потоков могут создаваться контроллером 921 коммутации согласно операциям по протоколу IFMP-C. Как упоминалось выше, контроллер 921 коммутации связан с коммутатором уровня канала (таким, как ATM коммутатор 925), который, в свою очередь, может быть связан с исполнительными устройствами коммутации (такими, как 91₁, 91₃, 91₅ и/или с другим системным узлом 91₆). В процессе инициализации контроллер 921 коммутации посылает пакеты протокола IFMP-C к исполнительным устройствам коммутации, что обеспечивает возможность контроллеру 921 коммутации определить конкретную конфигурацию (в смысле установленных сетевых интерфейсов и т.п.) каждого исполнительного устройства коммутации. Контроллер 921 коммутации затем кондиционирует один или более сетевых интерфейсов 91₇_n, связанных с исполнительными устройствами коммутации, для начала приема пакетов. Контроллер 921 коммутации также устанавливает пакеты обработки в исполнительных устройствах коммутации для передачи определенных принимаемых пакетов к контроллеру 921 коммутации, в то время как другие принимаемые пакеты могут быть опущены по протоколам, не обрабатываемым контроллером 921 коммутации). Если контроллер 921 коммутации обнаруживает, что поток может обрабатываться с помощью исполнительного устройства коммутации без вмешательства контроллера 921 коммутации, то контроллер 921 коммутации использует протокол IFMP-C для управления исполнительным устройством коммутации при обработке такого пакета (например, пропуске пакета, пересылке пакета с одного или более интерфейсов с использованием одного или более выходов форматов или с использованием других классов обслуживания для локальной пересылки пакета). С пересылкой пакета связано преобразование, применяемое для пакета (например, уменьшение "времени жизни" в пакете, обновление контрольной суммы заголовка IP, манипулирование заголовком для форматов других типов потока и т.п.). Ниже описаны дополнительные детали, относящиеся к взаимодействию исполнительных устройств коммутации, узлов коммутации и контроллера коммутации (например, как показано в конфигурации по фиг. 2с).

В соответствии с настоящим изобретением система добавляет полные

маршрутизации по протоколу IP к аппаратным средствам ATM коммутации (либо альтернативной технологии в других вариантах) с использованием системного программного обеспечения вместо любого имеющегося программного обеспечения управления ATM коммутатором, для управления ATM коммутатором. Таким образом, настоящая система способна осуществлять маршрутизацию по протоколу IP между сетевыми уровнями, если необходимо, и коммутацию уровней каналов передачи данных, если возможно, для создания высокоскоростной и высокопроизводительной локальной передачи эффективном способом, избегая проблем перегрузки при маршрутизации.

С использованием протокола управления потоком (rsion (IFMP), который детально описан ниже, системное программное обеспечение дает возможность системному узлу (например, базовому коммутационному узлу, шлюзовому узлу коммутации или главному компьютеру/серверу/рабочей станции) классифицировать пакеты протокола IP как принадлежащие к потоку сходных пакетов, основываясь на определенных общих характеристиках. Поток представляет собой последовательность пакетов, передаваемых от конкретного источника к конкретному адресу (одиночному или групповому), которые связаны в смысле их маршрутизации или иной локальной обработки, которая может потребоваться. Настоящее изобретение обеспечивает возможность различной обработки различных типов пакетов в зависимости от типа пакета.

Некоторые типы пакетов могут обрабатываться путем отбраковки их в индивидуальные ATM соединения с использованием ATM коммутатора для выполнения высокоскоростной коммутации пакетов. Поток, например, такие как несущие трафик реального времени, с требованиями качества обслуживания или требующие длительного времени выдержки, могут конфигурироваться для коммутации при наличии возможности. Другие типы потоков, например потоки короткой длительности или запросы в базы данных, обрабатываются путем IP маршрутизации без установления соединений. Конкретный поток пакетов может связываться с конкретной ATM меткой (т.е. ATM идентификатором виртуального маршрута (VPI) и идентификатором виртуального канала (VCII)). Предполагается, что виртуальные каналы являются одноплавающими, так что ATM метка входящего направления каждого канала принадлежит входному порту, с которым он соединен. Каждое направление передачи канала обрабатывается отдельно.

Разумеется, потоки, перемежающиеся в каждом направлении, обрабатываются системой отдельно аналогичным способом. Классификация потока представляет собой решение, принимаемое в соответствии с локальной спецификой. Когда пакет протокола IP принимается системным узлом, системный узел передает пакет протокола IP по каналу, устанавливаемому по умолчанию. Узел также классифицирует пакет протокола IP, как принадлежащий к конкретному потоку, и соответственно принимает решение.

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

поток, предопределенным образом коммутироваться непосредственно в ATM коммутатор, или они должны продолжаться пересылаться посылке с использованием программного обеспечения маршрутизации в узле. Если принято решение о коммутации потока пакета, то поток должен сначала маркироваться. Для маркировки сначала выбирает для этого потока доступную метку (VP/VC) входного порта, на который был принят пакет. Узел, который принял решение о маркировке потока, затем заполняет метку, идентификатор потока и время жизни и затем передает сообщение IFMP REDIRECT (перенаправление) по восходящей линии к предыдущему узлу, от которого поступил пакет. Идентификатор потока содержит набор полей заголовка, которые характеризуют поток. Время жизни определяет длительность времени, в течение которого действует перенаправление. Если только состояние потока не было обновлено, связь между потоком и меткой удаляется по истечению указанного времени жизни. Истечение времени жизни для обновления состояния потока приводит к тому, что последующие пакеты, принадлежащие потоку, будут передаваться по умолчанию, между установленным узлом, к каналу передачи, сообщаясь узлам. Состояние потока обновляется передачей по восходящей линии сообщения "перенаправление", имеющего те же самые метку и идентификатор потока, что и первоначальное, но имеющего другой время жизни. Сообщение "перенаправление" запрашивает узел восходящей линии передавать все последующие пакеты, которые имеют характеристики, совпадающие с характеристиками, указанными в идентификаторе потока, по виртуальному каналу, указанному меткой. Решение о перенаправлении представляет собой локальное решение, обрабатываемое узлом нисходящей линии. Соответственно даже если узел нисходящей линии запрашивает перенаправление конкретного потока пакетов, узел восходящей линии может принять решение о принятии или игнорировании запроса на перенаправление. Кроме того, для сообщения "перенаправление" не передается подтверждение приема. Напротив, первый пакет, приходящий по новому виртуальному каналу, служит для индикации того, что запрос перенаправления принят.

Системное программное обеспечение также использует различные процедуры формирования пакетов протокола IP, которые принадлежат к маршрутированному потоку в линии передачи ATM данных, в зависимости от различных типов потоков. В рассматриваемом варианте осуществления используются четыре типа формирования пакетов.

Помимо протокола IFMP, системное программное обеспечение использует другой протокол - общий протокол управления коммутацией (GSMР) - для установления связи по линии передачи ATM данных между контроллером коммутации и ATM коммутатором базового коммутиационного блока системы и при этом обеспечивая возможность коммутации на уровне 2, если возможно, и маршрутизации по протоколу IP на уровне 3 и пересылки пакетов, если

представляет собой универсальный асимметричный протокол управления ATM коммутатором. Это означает, что контроллер коммутации действует в качестве ведущего, в ATM коммутатор - в качестве подчиненного. Протокол GSMР реализуется в виртуальном канале, установленном при инициализации в линии передачи ATM данных между контроллером коммутации и ATM коммутатором. Один контроллер коммутации может использовать множество коммутиационных понятий GSMР по различным виртуальным каналам для управления множеством ATM коммутаторов. В протокол GSMР также включен протокол близости GSMР, который используется для синхронизации состояния в линии передачи ATM данных между контроллером коммутации и ATM коммутатором, для определения данных идентификации узла на другом конце линии связи и для обнаружения изменений в идентификации этого узла.

Протокол GSMР позволяет контроллеру коммутации устанавливать и освобождать соединения через ATM коммутатор, добавляя и исключая "листья" (элементы нижнего уровня иерархии в древовидной структуре иерархии) в соединения типа "от точки к множеству точек", управлять портами коммутации, запрашивать информацию конфигурации, и запрашивать статистику. Протокол GSMР также позволяет ATM коммутатору информировать контроллер коммутации о событиях, например отключении линии связи.

Предполагается, что коммутатор имеет множество портов, причем каждый порт представляет собой комбинацию входного и выходного портов. Элементы ATM данных (cell) поступают от внешней линии связи по входящим виртуальным каналам на входной порт и передаются от ATM коммутатора во внешнюю линию связи по исходящим виртуальным каналам с выходного порта. Как упоминалось выше, виртуальные каналы в порте или линии связи идентифицируются по их меткам виртуальных маршрутов и используются для виртуальных соединений. Виртуальные каналы в соединении входящего виртуального канала (или корня) с одним или более исходящими виртуальными каналами (или ветвями). Соединения с использованием виртуальных каналов идентифицируются по входному порту, на который они приходят, и по метке VP/VC из входящего виртуального канала. В коммутаторе каждый порт имеет таблицу периодов времени аппаратурных средств, индексированную метками VP/VC входящих ATM элементов данных, причем запись в таблице контролируется процессором.

В случае протокола GSMР каждое соединение с использованием виртуального канала может быть установлено с определенным качеством обслуживания, путем присвоения приоритета при его установлении. Для соединений с использованием виртуальных каналов совместно использующих один и тот же выходной порт, ATM элемент данных в соединении с более высоким приоритетом будет передан коммутатором с более высокой

соединении с более низким приоритетом, если они оба присутствуют в коммутаторе в одно и то же время. Цисро приоритетов, поддерживаемых каждым портом коммутатора, может быть определено из сообщения конфигурирования порта. Ясно, что различные коммутаторы могут поддерживать групповую передачу различными путями. Например, коммутатор может иметь пределы по числу ветвей для реализации группового соединения, пределы по количеству поддерживаемых групповых соединений, пределы по количеству различных значений VP/VC, которые могут быть присвоены выходным ветвям группового соединения, или поддерживать только одну ветвь конкретного группового соединения на одном и том же выходном порте. Если необходимо, могут быть определены коды отказа для соответствующих ситуаций.

Коммутатор присваивает 32-битовые номера портов для описания портов коммутатора. Номер порта может быть структурирован для включения субпортов, относящихся к физической структуре коммутатора (например, стойка, слот, порт). Каждый порт коммутатора также поддерживает номер сервиса связи порта, присвоенный коммутатором. Номер сервиса связи порта остается тем же самым при постоянном изменении состояния порта. Однако если порт возвращается в рабочее состояние после его отключения или недоступности или числа подчиненных, номер сервиса связи порта будет изменен. Номер сервиса связи порта присваивается с использованием некоторой формы случайного числа и позволяют контроллеру коммутации обнаруживать отказы линии связи и поддерживать состояние синхронизации.

Помимо протоколов IFMP и GSMР системное программное обеспечение в некоторых вариантах осуществления изобретения также использует еще один протокол - протокол управления потоком (flow) для клиентов (IFMP-C), который описан детально ниже, для установления информационного обмена по линии связи между контроллером коммутации базового коммутиационного блока и исполнительным устройством коммутации, чтобы распределять пересылку пакетов уровня 3 к исполнительным устройствам коммутации, если это желательно. В частности, протокол GFMP-C представляет собой универсальный асимметричный протокол для управления исполнительным устройством коммутации. Тем же образом, контроллер коммутации действует в качестве ведущего узла, а исполнительное устройство коммутации - в качестве подчиненного. При использовании протокола IFMP-C интерфейсы на исполнительном устройстве коммутации сходны с интерфейсами, присвоенными контроллеру коммутации, так что контроллер коммутации/исполнительное устройство коммутации с внешней стороны представляет подобный системному узлу. В принципе протокол IFMP-C используется в виртуальном канале, установленном при инициализации в линии связи между контроллером коммутации и исполнительным устройством коммутации. Один контроллер коммутации может использовать множество

отдельных виртуальных каналов для управления множеством исполнительных устройств коммутации. При запуске системы контроллер коммутации запускает приемник протокола IFMP-C в каждом ATM интерфейсе (приемнику поставлен в соответствие устанавливаемый по умолчанию виртуальный канал ATM интерфейса), связанном с контроллером коммутации, и исполнительное устройство коммутации начинает передачу периодических сообщений синхронизации (SYN) по установленному по умолчанию виртуальному каналу. Когда контроллер коммутации принимает сообщение SYN от исполнительного устройства коммутации, контроллер коммутации запускает протокол близости IFMP-C, который включен в протокол IFMP-C. Предназначенный для использования на каждом конце линии связи протокол близости IFMP-C используется для синхронизации состояния в линии связи между контроллером коммутации и исполнительным устройством коммутации, чтобы обнаружить идентификационные данные узла на другом конце линии связи и обнаружить изменения в идентификации этого узла. После того как протокол близости IFMP-C установит, что каждая сторона линии связи синхронизирована с другой стороной, каждая сторона линии связи будет иметь номер экземпляра записи, идентифицирующий другую сторону линии связи.

После завершения синхронизации протокол IFMP-C позволяет контроллеру коммутации определить, какие порты или интерфейсы (и их атрибуты) доступны в исполнительном устройстве коммутации, и конфигурировать каждый интерфейс таким образом, что он может быть использован для пересылки пакетов. Исполнительное устройство коммутации содержит множество портов или интерфейсов, причем каждый интерфейс или порт является комбинацией входного порта и выходного порта. После того как интерфейсы определены и конфигурированы, протокол IFMP-C используется для создания, модифицирования и исключения ветвей передачи. Каждая ветвь передачи состоит из входных данных и выходных данных. В каждом исполнительном устройстве коммутации каждый интерфейс имеет таблицу периодов времени аппаратурных средств, индексированную входными данными/выходными данными входящего пакета, в запись в таблице контролируется локальным процессором управления в исполнительном устройстве коммутации. Входные данные включаются в себя различные элементы и компоненты информации (например, входной интерфейс, предшествование, входные флаги, данные ключа, маску ключа в соответствии с конкретным вариантом осуществления), причем каждый элемент информации вносит вклад во входную информацию. Если какие-либо компоненты входных данных меняются, то пакет рассматривается как имеющий другую входную запись пересылки. Входные данные включают в себя различные элементы и компоненты информации (например, выходной интерфейс, длину/удаления, преобразование,

качество типа обслуживания, данные качества обслуживания в соответствии с конкретным вариантом осуществления), которые описывают то, каким образом должны пересылаться пакеты, имеющие согласованные входные данные. Возможно, что одна входная запись будет иметь более одной выходной записи. Когда пакет приходит на интерфейс исполнительного устройства коммуникации, исполнительное устройство коммуникации осуществляет поиск во входных записях, связанных с входным идентификатором.

Поиск записей осуществляется от низших к высшим по старшинству. Если обнаружено совпадение входных записей, то информация о выходных ветвях используется для пересылки пакета.

В случае протокола IGMP-S управление аппаратными средствами на уровне линии связи (например, открытие виртуальных каналов и добавление фильтров адресов аппаратных средств для сети, Ethernet) сохраняется за исполнительным устройством коммуникации. Если входная масса ключа включает биты адреса на уровне линии связи, то исполнительное устройство коммуникации должно гарантировать прием этих адресов. Если масса не включает информации адресации на уровне линии связи, то исполнительное устройство коммуникации не должно нестиравать фильтр. Исполнительное устройство коммуникации должно таким образом управлять фильтрацией на уровне линии связи наиболее эффективным образом для своих аппаратных средств, а контроллер коммуникации должен включать достаточное количество информации уровня линии связи в ключ, чтобы надлежащим образом осуществлять фильтрацию пакетов.

Контроллер коммуникации управляет состоянием исполнительного устройства коммуникации для разнородного и группового разнородного режимов передачи так, чтобы исполнительное устройство коммуникации не пыталось неадекватным образом оптимизировать кодировку маршрута вне пределов желательного режима.

Протоколы IGMP, GSNP и IGMP-S детально описаны ниже в соответствии с конкретным вариантом осуществления настоящего изобретения.

2. Аппаратные средства системы

На фиг.3 представлена блок-схема типовой компьютерной системы 51, которая может быть использована в качестве контроллера 5 коммуникации в базовом коммутационном блоке 1 (как показано на фиг.1в) для реализации системного программного обеспечения. На фиг.3 также показан пример компьютерной системы, которая может быть использована в качестве контроллера 23 шлюзового блока коммуникации в шлюзовом блоке 21 для реализации системного программного обеспечения, согласно способу настоящего изобретения, а также служащий примером типовой компьютерной системы, которая может быть использована в качестве главного компьютера/серверной станции с компьютером системного программного обеспечения. Разумеется, другие элементы, такие как монитор, экран,

компьютера. Как показано на фиг.3, компьютерная система 51 содержит такие подсистемы как центральный процессор 69, системная память 71, контроллер ввода/вывода 73, стационарный диск 79, сетевой интерфейс 81, ПЗУ 83. Компьютерная система 51 в случае, если она является главным компьютером ("хостом"), может дополнительно содержать контроллер 53 клавиатуры 59, адаптер дисплея 75, сетевой диск 77. Стрелки 65 представляют архитектуру системной шины компьютерной системы 51. Однако эти стрелки лишь иллюстрируют схему взаимных соединений, обеспечивающих связь между подсистемами. Например, локальная шина может использоваться для соединения центрального процессора 69 с системной памятью 71 и ПЗУ 83. Другие компьютерные системы, пригодные для использования в настоящем изобретении, могут включать дополнительные подсистемы или меньшее количество подсистем. Например, другая компьютерная система может включать более одного процессора 69 (мультипроцессорная система) или кэш-память (сверхоперативную память).

В возможном варианте осуществления изобретения компьютер, используемый в качестве контроллера коммуникации, представляет собой стандартную машину на базе центрального процессорного блока типа Intel[®], оснащенного стандартной шиной PCI (соединения периферийного оборудования), также адаптером ATM сети или платой сетевого интерфейса (NIC). Компьютер связан с ATM коммутатором посредством ATM линии связи со скоростью передачи 155 Мбит/с, использующей плату интерфейса ATM сети. В данном варианте осуществления системное программное обеспечение установлено на стационарном диске 79, который представляет собой накопитель на жестких дисках компьютера. Специалистам в данной области техники ясно, что системное программное обеспечение может храниться на ПЗУ на компакт-диске (CD-ROM), на гибком диске, на магнитной ленте или иных материальных носителях, которые могут хранить машиночитаемые коды.

Компьютерная система 51, показанная на фиг.3, представляет собой лишь возможный пример компьютерной системы, пригодной для использования в качестве контроллера для исполнения базового коммутационного блока, контроллера шлюзового блока коммуникации или главного компьютера/серверной станции) в соответствии с настоящим изобретением. Кроме того, фиг.3 иллюстрирует пример компьютерной системы, установленной с использованием по меньшей мере части системного программного обеспечения (для обеспечения режима работы в соответствии с протоколом IGMP-S), которая может быть использована в качестве исполнительного устройства коммуникации 901 (как показано на фиг.1с). Следует иметь в виду, что системное программное обеспечение для протоколов маршрутизации нет необходимости устанавливать в компьютерной системе, служащей в качестве исполнительного устройства 901 коммуникации, и поэтому данная часть системного программного обеспечения может реализовываться на встроеном

диск 79 может быть опущен в компьютерной системе, используемой в качестве исполнительного устройства 901 коммуникации, что приводит к меньшим затратам на аппаратуру для сетей, которые могут использовать исполнительные устройства 901 коммуникации вместо шлюзовых блоков коммуникации. Для специалистов в данной области техники очевидно, и другие конфигурации подсистем, пригодных для использования в соответствии с настоящим изобретением. Кроме того, шлюзовый блок коммуникации может быть оснащен множеством других плат сетевых интерфейсов (NIC) для обеспечения соединения с другими типами локальных сетей. Другие платы NIC или альтернативные адаптеры для других типов основных операционных сред могут быть использованы в шлюзовом блоке коммуникации. Например, могут использоваться платы SMC 10M/100M Ethernet или FDDI. Без каких-либо ограничений объема изобретения, табл. 1 представляет перечень коммерчески доступных компонентов, которые используются при эксплуатации контроллера в соответствии с вышеописанными вариантами осуществления. Для специалистов в данной области техники должно быть очевидно, что компоненты, приведенные в табл.1, являются лишь характеристиками компонентов для реализации изобретения и приведены для пояснения выполнения и придействия в соответствии с конкретным вариантом осуществления изобретения. Различные другие компоненты, известные специалистам в данной области техники, могут быть использованы взамен, либо в комбинации или отдельно с другими компонентами.

В Аппаратные средства коммуникации. Как описано выше, аппаратные средства коммуникации ATM предусматривают исполнительное устройство коммуникации или базовый коммутационный блок ATM коммутатор использовать независимые от поставщика аппаратные средства ATM коммуникации. Однако машина ATM коммуникации, соответствующая настоящему изобретению, не основывается на обычном, ориентированном на соединения программном обеспечении ATM маршрутизации и сигнализации (SSCOP, Q.2831, UNI 3.0/3.1, P-NNI). Напротив, протоколы и программное обеспечение режима ATM полностью не используются, а базовый коммутационный блок основывается на системном программном обеспечении для управления машиной ATM коммуникации. Системное программное обеспечение более детально описано ниже.

Отдельно доступные ATM компоненты могут быть собраны в типовой архитектуру ATM коммутатора. Например, на фиг.4 представлена обобщенная блок-схема архитектуры ATM коммутатора 3 (пример иллюстрирует 16-портовый коммутатор), который может быть использован в качестве аппаратного средства коммуникации базового коммутационного узла в соответствии с возможным вариантом осуществления изобретения. Однако коммерчески доступные ATM коммутаторы также могут работать в качестве машины коммуникации базового коммутационного блока в соответствии с

настоящего изобретения. Основные функциональные компоненты аппаратов средств коммуникации 3 включают переключатель сердечник, микроконтроллерный комплекс, переключательный подузел. В принципе, переключательный подузел выполняет 2 коммуникацию на уровне микроконтроллерного комплекса обеспечивая системное управление ATM коммутатором, и приемопередающий подузел обеспечивает интерфейс и базовую передачу и прием сигналов от физического уровня. В данном примере переключательный сердечник основан на комплексе микросхем ATM коммутатора MMC Networks ATMS 2000, который включает в себя "белую микросхему" 100, "серую микросхему" 102, микросхему MBUF 104, микросхему интерфейса порта (PIF) 106 и общую память данных 108. Переключательный сердечник может дополнительно включать дататор активности виртуального канала 110, функциональный блок 112 отбрасывания раннего пакета. Также включены счетчики пакетов, не показанные на чертеже. Белая микросхема 100 обеспечивает управление конфигурацией и статус. Помимо обмена данными с белой микросхемой 100, микросхема 102 обеспечивает прямую адресацию и перенос данных с использованием таблиц коммуникации. Микросхемы MBUF 104 обеспечивают переключение трафика элементов данных между микросхемами интерфейса порта 106 и памятью общих данных 108. Память общих данных 108 используется для буферизации элементов данных в коммутаторе. Микросхемы интерфейса порта 106 управляют переносом данных между микросхемами MBUF и аппаратными средствами портов коммуникации. Детектор активности 110 виртуального канала, включенный в себя элемент памяти, обеспечивает информацию о каждом действующем виртуальном канале. Функциональный блок 112 отбрасывания ранних пакетов обеспечивает возможность отбрасывать некоторые ATM элементы данных по мере необходимости. Счетчики пакетов обеспечивают в коммутаторе средство подсчета всех пакетов, проходящих через все входные и выходные порты. Шины 114, 115, 117 и 118 обеспечивают интерфейс между различными компонентами коммутатора. Микроконтроллерный комплекс включает в себя центральный процессорный блок 130, динамическую память со случайным доступом 132, ПЗУ 134, флэш-память 136, контроллер 138 динамической памяти со случайным доступом, сдвоенные универсальные входы/выходы, примитивизирующие порты 140 и 142 и внешний таймер 144. Центральный процессорный блок 130 действует в качестве микроконтроллера. ПЗУ 134 действует в качестве ПЗУ локальной самозагрузки и включает в себя полностью отображаемые коды переклочения, функциональные средства операционной системы базового нижнего уровня и диагностику. Динамическая память со случайным доступом 132 обеспечивает обычные функции ОЗУ, а контроллер 138 (который может быть

"переваривания" протокола IFMP по восходящей линии на этапе 200. После этапа 200 системный узел возвращается к этапу 176, как показано на фиг.5b. Фиг. 6b представляет диаграмму, иллюстрирующую этап, осуществляемый при коммутации потока в базовом коммутационном блоке, как показано этапом 180 коммутации потока на фиг.5b. Как упомянуто выше, только системные узлы, представляющие собой базовые коммутационные блоки, могут выполнять этап коммутации потока. Когда на этапе 210 начинается процедура коммутации потока, контроллер коммутации в базовом коммутационном блоке передает на этапе 212 сообщение протокола GSNP для отобранных меток х в восходящей линии связи в метку у в нисходящей линии связи. Метка у представляет собой метку, которую узел, размещенный в нисходящей линии связи, относительно базового коммутационного блока, присвоил потоку. Разумеется, этот узел нисходящей линии маркировал поток таким образом, как рассмотрено со ссылками на фиг.5b и 6a, причем свободная метка у выбирается на этапе 192. После этапа 212 контроллер коммутации в базовом коммутационном блоке ожидает на этапе 214 сообщения подтверждения протокола GSNP от ответного средства коммутации в базовом коммутационном блоке, осуществляющего, что отображение коммутации успешно. Поток при этом осуществляется на уровне 2 полностью аппаратным средством коммутации в базовом коммутационном блоке. Затем базовый коммутационный блок осуществляет переключку пакета на этапе 176.

На фиг.5c представлена диаграмма, иллюстрирующая этап, выполняемый при переключке пакета в системном узле так, как показано для этапа 176 переключки пакета на фиг.5b. Системный узел на этапе 218 запускает процедуру переключки пакет, в котором, к которому принадлежит пакет, не маркирован в нисходящей линии (этап 220), то системный узел передает пакет по виртуальному каналу, установленному по умолчанию в нисходящую линию на этапе 222 и затем переходит к состоянию ожидания 224 для ожидания поступления пакетов. Однако если поток, к которому принадлежит пакет маркирован в нисходящей линии, указывающей, что системный узел ранее принял сообщение "переваривания" протокола IFMP для маркировки этого потока во время жизни, то системный узел на этапе 226 проверяет, не истекло ли время жизни для переваривания данного пакета. Если время жизни на истекло, то системный узел передает пакет по маркированному виртуальному каналу в сообщении "переваривания" протокола IFMP на этапе 228, затем переходит в состояние ожидания 224. Если время жизни истекло, то системный узел автоматически исключает переваривание потока на этапе 230. Системный узел затем передает пакет по каналу, установленному по умолчанию (этап 222) и возвращается в состояние ожидания на этапе 182, как показано на фиг.5b.

Как описано выше, фиг.6a-6c в общем относятся к взаимодействию системных узлов

исполнительных устройств коммутации. Фиг. 6d-6e относятся к взаимодействию узлов коммутации, когда по меньшей мере одно исполнительное устройство коммутации связано с базовым коммутационным блоком, как описано ниже. На фиг. 6d представлена диаграмма, иллюстрирующая этап, выполняемый в контроллере коммутации при маркировке потока для пакетов, принимаемых от связанного исполнительного устройства коммутации, являющегося источником, как показано этапом 1660 маркировки потока на фиг.6d. На фиг.6d показаны три сценария: когда поток пакетов желательно передать на другой интерфейс исполнительного устройства коммутации, являющегося источником; когда поток пакетов желательно передать на интерфейс другого связанного исполнительного устройства коммутации, т.е. на исполнительное устройство коммутации, являющееся адресатом; и когда поток пакетов желательно передать на интерфейс другого связанного системного узла (или узла коммутации, такого как другой базовый коммутационный блок, шлюзовой блок коммутации или главный компьютер).

Как показано на фиг.6d, если поток пакетов, принятый от исполнительного устройства коммутации, являющегося источником, желательно передать на другой интерфейс того же самого исполнительного устройства коммутации (как определено на этапе 1662), контроллер коммутации (на этапе 1664) использует протокол IFMP-C для кондиционирования исполнительного устройства коммутации, являющегося источником, чтобы направлять последующие пакеты, принимаемые для данного потока с соответствующим заголовком и преобразованием в интерфейс адресата данного исполнительного устройства коммутации.

Если поток пакетов, принимаемых от исполнительного устройства коммутации, являющегося источником, нежелательно направлять к другому интерфейсу того же самого исполнительного устройства коммутации (как определено на этапе 1662), то на этапе 1666 определяется, не следует ли направлять поток пакетов, принимаемых от исполнительного устройства коммутации, являющегося источником, к интерфейсу исполнительного устройства коммутации, являющегося адресатом. Если это так, то контроллер коммутации (на этапе 1668) выбирает свободную метку х в восходящей линии связи между контроллером коммутации и исполнительным устройством коммутации, являющимся источником, и выбирает (на этапе 1670) свободную метку у в нисходящей линии связи между контроллером коммутации и исполнительным устройством коммутации, являющимся адресатом. Затем контроллер коммутации использует протокол GSNP для отобранных х и у на этапе 1672. На этапе 1674 контроллер коммутации использует протокол IFMP-C для кондиционирования исполнительного устройства коммутации, являющегося адресатом, чтобы направлять на интерфейс адресата последующие пакеты, принимаемые с меткой у с соответствующим заголовком и преобразованием. Затем на этапе 1676 контроллер коммутации

кондиционирования исполнительного устройства коммутации, являющегося источником, чтобы направлять последующие пакеты потока с соответствующим заголовком и преобразованием к метке х. Если поток пакетов, принимаемых от исполнительного устройства коммутации, являющегося источником, нежелательно передавать на интерфейс исполнительного устройства коммутации, являющегося адресатом (что определено на этапе 1666), то поток пакетов, принимаемых от исполнительного устройства коммутации, являющегося источником, желательно передать на интерфейс другого связанного системного узла (или узла коммутации, такого как другой базовый коммутационный блок, шлюзовой блок коммутации или главный компьютер). В этом случае контроллер коммутации (на этапе 1680) выбирает свободную метку х в восходящей линии между контроллером коммутации и исполнительным устройством коммутации, являющимся источником. На этапе 1682 контроллер коммутации ожидает свободной метки у в нисходящей линии, выбираемой коммутационным узлом и пересылаемой посредством протокола IFMP. Затем контроллер коммутации использует протокол GSNP для отобранных х и у на этапе 1684. На этапе 1686 контроллер коммутации использует протокол IFMP-C для кондиционирования исполнительного устройства коммутации, являющегося источником, для переключения последующих пакетов потока с соответствующим заголовком и преобразованием к метке х.

На фиг. 6e представлена диаграмма, иллюстрирующая этап, выполняемый в контроллере коммутации при маркировке потока (начиная с этапа 1700) для пакетов, которые принимаются от связанного коммутационного узла и предлагаются для интерфейса на связанном исполнительном устройстве коммутации. Если поток пакетов, принимаемых от коммутационного узла, являющегося источником, желательно направлять к интерфейсу исполнительного устройства коммутации, являющегося адресатом, то контроллер коммутации (на этапе 1702) выбирает свободную метку х в восходящей линии между контроллером коммутации и коммутационным узлом, являющимся источником, и выбирает (на этапе 1704) свободную метку у в нисходящей линии между контроллером коммутации и исполнительным устройством коммутации, являющимся адресатом. Затем контроллер коммутации использует протокол GSNP для отобранных х и у на этапе 1706. На этапе 1708 контроллер коммутации использует протокол IFMP-C для приваждения исполнительного устройства коммутации адресата в состояние для направления в интерфейс адресата последующих пакетов от потока, принимаемого по метке у с соответствующим заголовком и преобразованием. На этапе 1710 контроллер коммутации использует протокол IFMP для запроса коммутационного узла восходящей линии от передаче последующих пакетов потока к метке х. Дополнительные детали приведенного выше описания представлены ниже.

А. Протокол IFMP и передача

АТМ данных 1. Протокол IFMP Системное программное обеспечение использует протокол управления потоком Ipsilon (IFMP) для обеспечения возможности системному узлу (такому, как базовый коммутационный блок, шлюзовой блок коммутации или главный компьютер/сервер/рабочая станция) классифицировать пакеты протокола IP как принадлежащие к потоку стодных пакетов, основываясь на некоторых общих характеристиках. Потом определяется "идентификатор потока". Идентификатор потока для конкретного потока дает содержимое или значения набора полей из заголовка пакета, которые определяют поток. Содержимое набора полей из заголовков пакетов является одним и тем же для всех пакетов, принадлежащих данному конкретному потоку. Могут быть определены различные типы пакетов. Каждый тип потока определяет набор полей из заголовка пакета, которые используются для определения потока. Например, один тип потока может определять набор полей из заголовка пакета, который определяет поток как миксидый пакет, переносимые данные между приложениями, задачами, реализованными на станциях, в то время как другой тип пакета может определять набор полей из заголовка пакета, который определяет поток как миксидый пакет, переносимые данные между станциями.

В одном из вариантов настоящего изобретения определены три типа потоков: тип 0, поток типа 1 и поток типа 2. Разумеется, могут быть определены и другие или дополнительные типы потоков. Поток типа 0 используется для парковки на формирование пакетов по протоколу IP от формирования пакетов по умолчанию. Формат идентификатора потока для потока типа 0 является нулевым и соответственно имеет нулевую длину. Тип 1 представляет собой тип потока, который определяет набор полей из заголовка пакета, идентифицирующий поток как содержащий пакеты, переносимые данные между приложениями задачами, реализованными на станциях. Тип 1 потока полезно использовать для потоков, содержащих пакеты для дайтеграммы пользователя) и TCP (протокол управления передачей), в которых первые четыре байта после IP-заголовка определяют номер порта источника и номер порта адресата, которые используются для указания исполняемой прикладной задачи. Идентификатор потока для типа 1 потока имеет длину четырех 32-битовых слов. Формат идентификатора потока для типа 1 потока, указанный со ссылкой на позицию 240 на фиг. 7a, включает (в порядке от старшего бита к младшему биту) версию, длину Интернет-заголовка (IHL), тип обслуживания, время жизни и поля протокола в качестве первого слова; поле адреса источника в качестве второго слова; поле адреса места назначения (адресата) в качестве третьего слова. Эти поля в идентификаторе потока являются частью IP-пакета потока типа 1. Идентификатор потока для потока типа 1 также включает поля номера порта источника

байта в IP-пакета после IP-заголовка) в качестве четвертого слова. Тип 2 представляет собой тип потока, который идентифицирует набор полей из заголовка пакета, передающие данные между станциями баз определения прикладных задач, выполняемых на станциях. Идентификатор потока для типа 2 имеет длину трех 32-битовых слов, формирует идентификатор потока для типа 2 потока, указанного в позиции 250 на фиг.7b, включая версию, длину Интернет-заголовка (HL), тип обслуживания, время жизни, протокол, поля адреса источника и адреса места назначения (адреса) из заголовка IP-пакета. Формат идентификатора потока для потока типа 2 тот же самый, что и для потока типа 1, без четвертого слова. Иерархический характер идентификаторов потоков для разных типов потоков позволяет выполнять конкретные операции сравнения для IP-пакетов для облегчения классификации потоков.

Настоящее изобретение обеспечивает возможность обрабатывать различные типы потоков отдельно, в зависимости от типа потока. Поток, например, такие, как переносимые графические, содержащие требования к качеству обслуживания или имеющие длительные время обслуживания, могут быть конфигурированы так, чтобы коммутироваться по мере возможности. Другие типы потоков, такие, например, как поток короткой длительности или запросы в базы данных, обрабатываются с использованием пересылки пакетов по протоколу IP без установления соединений. Кроме того, каждый тип потока также определяет тип формирования пакета (инкапсуляция), имеющего ту же самую метку и идентификатор потока, что и исходные, и имеющего другое время жизни. Сообщение "перераздача" указывает метку последующей линии, передающей все характеристики, совпадающие с характеристиками, совпадающими с потоком, по виртуальному каналу, определенному меткой. Решение о перераздачи принимает собой также решение местного уровня, обрабатываемое узлом восходящей линии, в то время как решение о классификации потока является решением, принимаемым на местном уровне, обрабатываемым узлом нисходящей линии. Соответственно даже если узел нисходящей линии запросит перераздачу конкретного потока пакетов, узел восходящей линии может принять решение принять или игнорировать запрос о перераздаче. Кроме того, сообщения "перераздача" не подтверждаются. Вместо этого первый пакет, приходящий по новому виртуальному каналу, служит для указания на то, что запрос перераздачи принят.

В настоящем изобретении протокол IFMR системного программного обеспечения включает протокол близости IFMR и протокол перераздачи IFMR. Протокол близости IFMR позволяет системному узлу ("хосту" базовому коммутиационному блоку или шлюзовому блоку коммутиации) определить идентификационные данные системного узла мультитиплексирования канала передачи

Как описано выше, классификация потока является решением, принимаемым на местном уровне. Если IP-пакет принимается системным узлом, системный узел передает IP-пакет по каналу, установленному по умолчанию. Узел также классифицирует IP-пакет как принадлежащий к конкретному потоку, и соответственно принимает решение, надо ли последующие пакеты, принадлежащие тому же самому потоку, коммутировать непосредственно в ATM коммутатор или продолжать пересылать посылно с помощью программного обеспечения маршрутизатора в узле. Если принимается решение коммутировать поток пакетов, то узел выбирает для данного потока доступную метку (VPI/VCI) входного порта, на который был принят пакет. Узел, который принял решение коммутировать поток, затем записывает метку, идентификатор потока и время жизни и передает сообщение "перераздача" (REDIRECT) протоколу IFMR в восходящую линию к предыдущему узлу, от которого поступил пакет. Как описано выше, идентификатор потока содержит набор полей времени, в течение которой действует перераздача. Если только состояние потока не обновляется, связь между потоком и меткой должна быть отменена после истечения времени жизни. Истечение времени жизни до момента обеспечения состояния приводит к тому, что последующие пакеты, принадлежащие к потоку, будут передаваться по каналу передачи, установленному по умолчанию между соседними узлами.

Состояние потока обновляется путем передачи в восходящую линию сообщения "перераздача", имеющего ту же самую метку и идентификатор потока, что и исходные, и имеющего другое время жизни. Сообщение "перераздача" указывает метку последующей линии, передающей все характеристики, совпадающие с характеристиками, совпадающими с потоком, по виртуальному каналу, определенному меткой. Решение о перераздаче принимает собой также решение местного уровня, обрабатываемое узлом восходящей линии, в то время как решение о классификации потока является решением, принимаемым на местном уровне, обрабатываемым узлом нисходящей линии. Соответственно даже если узел нисходящей линии запросит перераздачу конкретного потока пакетов, узел восходящей линии может принять решение принять или игнорировать запрос о перераздаче. Кроме того, сообщения "перераздача" не подтверждаются. Вместо этого первый пакет, приходящий по новому виртуальному каналу, служит для указания на то, что запрос перераздачи принят.

В настоящем изобретении протокол IFMR системного программного обеспечения включает протокол близости IFMR и протокол перераздачи IFMR. Протокол близости IFMR позволяет системному узлу ("хосту" базовому коммутиационному блоку или шлюзовому блоку коммутиации) определить идентификационные данные системного узла

протокол близости IFMR используется для синхронизации состояний в линии связи, для определения того, когда системный узел на другом конце линии связи изменится, и для обмена переносимых IP адресов, присвоенных линии связи. С использованием протокола перераздачи IFMR система может направлять сообщения "перераздача" по линии связи только после того, как система использует протокол близости IFMR для идентификации другого системного узла на другом конце линии связи и достижения синхронизации состояний в линии связи. Любое сообщение "перераздача", принимаемое по линии связи, не достигшей синхронизации состояний, должно обрабатываться. Протокол близости IFMR и протокол перераздачи IFMR детально описаны в изобретении ниже подробного описания работы системы.

Для иллюстрации преимущества, предоставляемых системой, будет рассмотрен конкретный пример, описывающий классификацию потока и перераздачу в соответствии с настоящим изобретением при использовании конфигурации локальной сети, подобной представленной на фиг.2a. В частности, в примере рассматривается взаимодействие между первым и вторым шлюзовыми блоками коммутиации 1 и базовым коммутиационным блоком 1 по фиг.2a. При запуске системы, выполняемых на контроллерах базового коммутиационного блока 1 и каждого из соседних узлов (в данном примере первого и второго шлюзовых блоков коммутиации 21), по умолчанию устанавливается виртуальный канал пересылки ATM данных. Когда IP-пакет передается от основной операционной среды 35, локальной сети по линии связи 39,1, сетевого уровня, IP-пакет принимается первым шлюзовым блоком коммутиации 21 последующей линии, передающей все характеристики, совпадающие с характеристиками, совпадающими с потоком, по виртуальному каналу, определенному меткой. Решение о перераздаче принимает собой также решение местного уровня, обрабатываемое узлом восходящей линии, в то время как решение о классификации потока является решением, принимаемым на местном уровне, обрабатываемым узлом нисходящей линии. Соответственно даже если узел нисходящей линии запросит перераздачу конкретного потока пакетов, узел восходящей линии может принять решение принять или игнорировать запрос о перераздаче. Кроме того, сообщения "перераздача" не подтверждаются. Вместо этого первый пакет, приходящий по новому виртуальному каналу, служит для указания на то, что запрос перераздачи принят.

В настоящем изобретении протокол IFMR системного программного обеспечения включает протокол близости IFMR и протокол перераздачи IFMR. Протокол близости IFMR позволяет системному узлу ("хосту" базовому коммутиационному блоку или шлюзовому блоку коммутиации) определить идентификационные данные системного узла мультитиплексирования канала передачи

продолжать маршрутизировать их потоком с использованием системного программного обеспечения. Если программное обеспечение контроллера коммутиации принимает решение, что поток должен коммутироваться, то оно выбирает свободную метку (метку X) из пространства меток (пространства меток) и предоставляет собой просто определенный диапазон идентификаторов (VPI/VCI) входного порта (порта 1), на котором принят пакет. Контроллер коммутиации также выбирает свободную метку (метку X) на своем порте управления (реальный или виртуальный порт, которым контроллер коммутиации соединен с ATM коммутатором). С использованием протокола GSMR системное

программное обеспечение подает команду ATM коммутатору отображать метку X на входном порте 1 в метку X' на порте управления с. После того как коммутатор возвратит назад сообщение подтверждения приема протокола GSMR, контроллер коммутиации направляет сообщение "перераздача" (IFMR REDIRECT) в восходящую линию предыдущему узлу (в данном примере первому шлюзовому узлу коммутиации 21), от которого поступил данный пакет. Сообщение "перераздача"

представляет собой просто команду от базового коммутиационного блока 1 к первому шлюзовому узлу коммутиации 21 передавать все последующие пакеты с полами заголовка, совпадающими с соответствующими полями, указанными в идентификаторе потока сообщения "перераздача" в виртуальном ATM канале, определенный меткой сообщения "перераздача". Если только состояние потока не будет обновлено, прежде чем истечет время жизни сообщения "перераздача", связь между потоком и меткой сообщения "перераздача" должна быть отменена, что приведет к тому, что последующие пакеты потока будут пересылаться по каналу, установленному по умолчанию (установленному первоначально при запуске) между базовым коммутиационным блоком 1 и первым шлюзовым блоком коммутиации 21.

Если первый шлюзовый блок коммутиации 21 принимает запрос в сообщении "перераздача", пересланном базовым коммутиационным блоком 1, пакеты, принадлежащие данному потоку, будут прибывать на порт контроллера коммутиации с меткой X' ATM VPI/VCI. Пакеты будут переконфигурированы и маршрутизированы системным программным обеспечением, но процедура решения о маршрутизации для пакета ускорена в результате того, что поток зашифровано и индексируется меткой X'. Соответственно очевидно, что поток может быть маркирован, но не обязательно должен коммутироваться.

Одно из важных преимуществ коммутиации становится очевидным в ситуации, когда узел нисходящей линии (в данном примере второй шлюзовый блок коммутиации) также участвует в перераздаче того же самого потока. Когда базовый коммутиационный блок 1 маршрутизирует первоначальный пакет, принадлежащий потоку, к второму шлюзовому блоку коммутиации 21 по установленному по умолчанию между ними каналу, то узел

устанавливается на значение поля 310 эземплера равного по положению из входящего сообщения, которое обусловлено формированием сообщения RSTACK. В сообщении SYN, SYNACK, ACK протокола близости IPMP поле 310 эземплера равного по положению представляет собой то, что передатчик считает текущим номером эземплера для равного по положению для данной линии. Если в передатчике не известен текущий номер эземплера для равного по положению для данной линии, то поле 310 эземплера равного по положению устанавливается в нуль. В сообщении RSTACK протокола близости IPMP поле 310 эземплера равного по положению устанавливается на значение поля 308 эземплера передатчика из входящего сообщения, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости IPMP поле 312 идентификации равного по положению представляет собой IP-адрес равного по положению, который, как представляется для данного передатчика сообщения, находится на другом конце линии связи. Передатчик принимает IP-адрес, который находится в поле адреса источника IP-заголовка, инкапсулирующего сообщение SYN или SYNACK, принятого передатчиком, и использует этот IP-адрес в поле 312 идентификации равного по положению в сообщении протокола IPMP, которое он передает. Если передатчик не знает IP-адрес равного по положению на другом конце линии связи, поле 312 идентификации равного по положению устанавливается на нуль. Для сообщения RSTACK протокола близости IPMP поле 312 идентификации равного по положению устанавливается на значение IP-адреса поля адреса источника из IP-заголовка входящего сообщения, которое вызвало генерирование сообщения RSTACK.

Поле 314 следующего номера последовательности равного по положению дает значения поля следующего номера последовательности равного по положению, который передатчик ожидает получить в следующем сообщении протокола пересдачи IPMP. Если значения поля 314 следующего номера последовательности равного по положению во входящем сообщении подтверждения (ACK) протокола близости IPMP больше, чем значение, равное единице плюс значение номера последовательности (из последнего сообщения протокола пересдачи IPMP, переданного с порте, на который было принято сообщение ACK протокола близости IPMP), то линия должна устанавливаться в исходное состояние.

Поле 316 перечня адресов представляет собой перечень из одного или более IP-адресов, которые присвоены линии IPMP. Перечень сообщений протокола близости IPMP. Перечень должен иметь по меньшей мере одну запись, которая идентифицирует источник IP-заголовка сообщения протокола близости IPMP. Содержимое перечня не используется протоколом IPMP, а может использоваться для протокола маршрутизации.

На фиг.8с представлена диаграмма,

после приема пакета с входящим сообщением протокола близости IPMP. После запуска системы системный узел принимает пакет с входящим сообщением протокола близости IPMP (этап 320). На этапе 322 системный узел определяет, является ли входящее сообщение протокола близости IPMP сообщением RSTACK. Если входящее сообщение протокола близости IPMP не является сообщением RSTACK (например, сообщение SYN, SYNACK, ACK), то системный узел работает в соответствии с тем, как показано на диаграмме состояния на фиг.8d. Если входящее сообщение протокола близости IPMP является сообщением RSTACK, то системный узел передает на этапе 324, совпадают ли значения эземплера передатчика и IP-адреса источника во входящем сообщении RSTACK со значениями, запомненными из предыдущего сообщения посредством операции обновления верификатора равного по положению. Для протокола близости IPMP операция обновления верификатора равного по положению определяется как запомненные значения передатчика и IP-адреса источника из сообщения SYN или SYNACK, принятого от равного по положению на конкретный порт. Если на этапе 324 установлено, что указанные значения совпадают, то системный узел на этапе 326, совпадают ли значения эземплера равного по положению и идентификации равного по положению во входящем сообщении RSTACK со значениями эземплера передатчика и IP-адреса источника, используемыми для всех сообщений SYN, SYNACK или ACK, передаваемых с порта, на который было принято входящее сообщение RSTACK. Если на этапе 326 установлено, что указанные значения совпадают, то системный узел на этапе 328 определяет, находится ли системный узел в состоянии IPMP. Если системный узел не находится в состоянии IPMP, то системный узел осуществляет SYNSENT. Если системный узел осуществляет SYNSENT, то системный узел осуществляет установку линии в исходное состояние на этапе 30. Если на этапе 324 или на этапе 326 установлено, что значения не совпадают, или системный узел не находится в состоянии IPMP, то системный узел на этапе 332 отправляет входящее сообщение RSTACK и ожидает прихода другого пакета. Соответственно, когда в системный узел приходит сообщение RSTACK протокола близости IPMP, системный узел устанавливает линию в исходное состояние, как показано этапами 334, 336, 338, 340 и 342. На этапе 344 системный узел генерирует новый номер эземплера для линии. Затем системный узел на этапе 336 отменяет верификатор равного по положению (т.е. устанавливает запомненные значения эземплера передатчика и IP-адреса источника для равного по положению в нуль). На этапе 338 системный узел устанавливает номер последовательности для следующего по номеру последовательности для равного по положению в нуль. Затем системный узел передает сообщение SYN протокола близости IPMP на этапе 340 и на этапе 342 входит в состояние IPMP. Системный узел затем принимает другой пакет для его обработки.

На фиг.8d представлена диаграмма

переходящего системного узла, когда входящее сообщение протокола близости IPMP не является сообщением RSTACK. Для последующего описания фиг.8d условие "B" определяется следующим образом: эземплер передатчика и IP-адрес источника во входящем сообщении совпадают со значениями, запомненными из предыдущего сообщения верификатора равного по положению для порта, на который было принято входящее сообщение протокола близости IPMP. Условие "C" на фиг.8d определяется следующим образом: эземплер равного по положению и идентификация равного по положению во входящем сообщении совпадают со значениями эземплера используемых в настоящее время для всех сообщений SYN, SYNACK, ACK, переданных с порта, на котором было принято входящее сообщение протокола близости IPMP. На фиг.8d условие "A" означает, что передающий системный узел принимает входящее сообщение SYNACK протокола близости IPMP, и что условие "B" удовлетворено. Условие "B" означает, что передающий системный узел принимает входящее сообщение ACK протокола близости IPMP, и что условия "B" и "C" удовлетворены. И условие "C" означает, что передающий системный узел принимает входящее сообщение ACK протокола близости IPMP, и что условия "B" и "C" не удовлетворены.

Если передатчик находится в состоянии 350 SYNSENT и принимает входящее сообщение SYN протокола близости IPMP от равного по положению на другом конце линии связи, то передатчик выполняет операцию обновления верификатора равного по положению и передает сообщение SYNACK протокола близости IPMP к равному по положению (показано как этап 352). Затем передатчик переходит из состояния 350 SYNSENT в состояние 354 SYNRCVD. Если SYN протокол близости IPMP, находясь в состоянии 354 SYNRCVD, то на этапе 352 передатчик выполняет операцию обновления верификатора равного по положению и передает сообщение SYNACK протокола близости IPMP к равному по положению, но остается в состоянии 354 SYNACK. Если передатчик находится в состоянии 354 SYNRCVD и либо состояние B, либо состояние D удовлетворено, то передатчик передает сообщение RSTACK протокола близости IPMP к равному по положению (показано как этап 356) и остается в состоянии 354 SYNRCVD. Если передатчик находится в состоянии 354 SYNRCVD и находится в состоянии 354 SYNRCVD и состояние C удовлетворено, то передатчик передает сообщение ACK протокола близости IPMP к равному по положению (показано как этап 358) и переходит в состояние 360 ESTAB. Если передатчик находится в состоянии 354 SYNRCVD и состояние A удовлетворено, то передатчик выполняет операцию идентификации обновления

передает сообщение ACK протокола близости IPMP к равному по положению (показано как этап 362) и переходит в состояние 360 ESTAB. Передатчик продолжает находиться в состоянии 360 ESTAB, если передатчик принимает сообщение SYN или SYNACK протокола близости IPMP, или если удовлетворено условие C. Если условие D удовлетворено, когда передатчик находится в состоянии 360 ESTAB, то передатчик остается в состоянии 360 ESTAB и передает сообщение RSTACK протокола близости IPMP (показано как этап 366). В состоянии SYNSENT, если либо передатчик принимает сообщение ACK протокола близости IPMP, либо удовлетворено условие B, то передатчик остается в состоянии 350 SYNSENT и передает сообщение RSTACK протокола близости IPMP (этап 356).

Если условие A удовлетворено, когда передатчик находится в состоянии 350 SYNSENT, то передатчик выполняет операцию обновления верификатора равного по положению и передает сообщение ACK протокола близости IPMP (этап 362) и переходит в состояние 360 ESTAB.

Как описано выше, протокол пересдачи IPMP используется для передачи сообщений пересдачи по линии связи, после того как система использовала протокол близости IPMP для идентификации другого системного узла на другом конце линии связи для достижения синхронизации состояний в линии связи. Любое сообщение пересдачи протокола IPMP, принятое в линии, которая еще не осуществила синхронизации состояний, должно быть отвергнуто. На фиг.9а представлена обобщенная структура сообщения 380 протокола пересдачи IPMP. Подобно всем сообщениям протокола близости IPMP, все сообщения протокола пересдачи IPMP инкапсулированы в IP-пакете. На фиг.9б показан обобщенный IP-пакет (в его современной версии IPv4) с переменной длиной поля данных, в котором может быть инкапсулировано сообщение протокола пересдачи. В качестве указания на то, что IP-пакет содержит сообщение протокола IPMP, поле протокола IP-заголовка инкапсулирующего IP-пакета должно содержать десятичное значение 101, а поле "трайма жизни" в заголовке IP-пакета инкапсулирующего сообщение протокола IPMP установлено в 1. Сообщение протокола пересдачи IPMP передается в IP-адрес равного по положению на другом конце линии (IP-адрес может быть получен из протокола близости IPMP с использованием IP-адреса в поле адреса места назначения IP-заголовка. Как показано на фиг.9а, сообщение 380 протокола пересдачи IPMP включает (в порядке от старшего бита к младшему): 8-битовое поле "версия" (382), 8-битовое поле "Op Code" (384), 16-битовое поле "контрольная сумма" (386) в качестве первого 32-битового слова; эземплер передатчика (388) в качестве второго 32-битового слова; эземплер "равного по положению" (390) в качестве третьего 32-битового слова; номер последовательности (392) в качестве четвертого 32-битового слова и тип сообщения (394), представляющего собой поле переменной длины из 32-битовых слов.

IFMP поле "версия" 392 определяет версию протокола IFMP, используемую в данный момент времени. Поле "Op Code" 394 определяет функцию протокола перерасдачи IFMP. В расширенном варианте осуществления имеются пять возможных кодов, т.е. функции сообщений протокола перерасдачи IFMP: REDIRECT (сообщение перерасдачи потока, Op Code=4), RECLAIM (сообщение восстановления метки, Op Code=5), RECLAIM ACK (сообщение подтверждения восстановления метки, Op Code=6), LABEL_RANGE (сообщение диапазона меток, Op Code=7) и ERROR (сообщение ошибки, Op Code=8).

Контрольная сумма 385 представляет собой 16-битовое дополнение до единицы суммы дополнений до единицы следующих величин: полей адреса источника, адреса места назначения, протокола из IP-пакета, инкапсулирующего сообщение протокола перерасдачи IFMP, и общей длины сообщения протокола перерасдачи IFMP. Контрольная сумма 385 используется системой для целей контроля ошибок.

В сообщении протокола перерасдачи IFMP элемент перерасдачи 388 представляет собой номер экземпляра перерасдачи для линии, полученный из протокола близости IFMP. В сообщении протокола перерасдачи IFMP поле 390 элемента равного по положению к элементу равно, что, как представляется в перерасчете, является текущим номером экземпляра равного по положению для линии связи, из-за этого определено из протокола близости IFMP.

Поле 392 номера последовательности послеплат системному узлу, принимающему сообщение протокола перерасдачи IFMP, обрабатывать сообщение протокола перерасдачи IFMP по порядку. Номер последовательности 392 получают приращение на единицу, по модулю 2³², на каждое сообщение протокола перерасдачи IFMP, передаваемое по линии связи. Номер близости IFMP устанавливает порядок последовательности в нуль при установлении связи в исходное состояние.

Поле 316 тела сообщения содержит перечень одного или более элементов сообщения протокола перерасдачи IFMP. Все элементы сообщения в перечне имеют один и тот же тип сообщения, так как поле 384 "Op Code" применимо ко всему сообщению протокола перерасдачи IFMP. Число элементов сообщения, включенных в один пакет, не должно приводить к тому, что суммарный размер сообщения протокола перерасдачи IFMP превышает максимальный размер передаваемого блока (MTU) для данной линии передачи данных. Для сообщений "диапазон меток" и "ошибка" протокола перерасдачи IFMP используется один элемент сообщения.

На фиг.90 представлена обобщенная диаграмма, иллюстрирующая работу системного узла после приема сообщения протокола перерасдачи IFMP. После запуска системный узел на этапе 400 принимает пакет, инкапсулирующий сообщение протокола перерасдачи IFMP. На этапе 402 системный узел проверяет, достигнута ли протоколом близости IFMP

Если синхронизация состояний не достигнута, системный узел отбрасывает пакет, инкапсулирующий принятый сообщение протокола перерасдачи IFMP (показано этапом 404). Если синхронизация состояний достигнута, то системный узел на этапе 406 проверяет IP-адрес источника, экземпляр перерасдачи 388 и экземпляр равного по положению 390 для пакета сообщения протокола перерасдачи IFMP. Если системный узел на этапе 408 определяет, что поля экземпляра перерасдачи 388 и IP-адреса источника входящего сообщения протокола перерасдачи IFMP не совпадают со значениями, запомненными в операции обновления верификатора равного по положению протокола близости IFMP для порта, на котором принято входящее сообщение протокола перерасдачи IFMP, то системный узел отбрасывает пакет входящего сообщения протокола перерасдачи IFMP (этап 404). Если на этапе 408 обнаружено, что указанные значения совпадают, то системный узел на этапе 410 определяет, совпадает ли поле 390 экземпляра равного по положению к текущему значению для экземпляра перерасдачи протокола близости IFMP. Если на этапе 408 определено, что указанные значения не совпадают, то системный узел отбрасывает пакет (этап 404). Однако если на этапе 408 определено, что эти значения совпадают, то системный узел продолжает (412) обрабатывать принятые сообщения протокола перерасдачи IFMP в соответствии с тем, как это необходимо.

Как описано выше, сообщение протокола перерасдачи IFMP может представлять собой сообщение REDIRECT перерасдачи, которое используется для подачи команды соседнему узлу присвоить одну или более меток пакетам, принадлежащим одному или нескольким конкретным потокам, причем для каждого в течение определенного времени. Системный узел, принимающий сообщение REDIRECT от узла, находящейся линии, принимает решение, следует ли принять запрос перерасдачи, и сообщать к перерасдачи. REDIRECT, и сообщать REDIRECT на подтверждается обменным способом.

Вместо этого реальная перерасдача пакетов с присвоенными метками для конкретных потоков вызывает, что системный узел принял запрос перерасдачи, содержащийся в сообщении REDIRECT. Каждый элемент сообщения REDIRECT в теле сообщения 394 сообщения IFMP имеет структуру, показанную на фиг.9с. Элемент 420 сообщения REDIRECT включает в себя (в порядке от старшего бита к младшему) следующие поля: 8-битовое поле 422 типа потока, 8-битовое поле 424 длины идентификатора потока, 16-битовое поле 426 времени жизни в составе первого 32-битового слова, 32-битовое поле 428 метки в качестве второго 32-битового слова, идентификатор потока 430, который представляет собой поле из целого кратного 32-битовых слов. Поле 422 типа потока определяет тип потока идентификатора потока, содержащегося в поле 430 идентификатора потока, в поле 424 длины идентификатора потока определяет длину

кратных 32-битового слова. Поле 425 времени жизни определяет длительность времени (в секундах), в течение которой действует перерасдача. Как описано в общем виде выше, после истечения интервала времени, определенного в поле 425 времени жизни, связь между идентификатором потока и меткой для конкретного потока, формат которой зависит от типа физической линии связи, по которой пересылается сообщение протокола перерасдачи IFMP. Поле 430 идентификатора потока идентифицирует поток, с которым связана конкретная метка в поле 428 метки.

В элементах сообщения протокола перерасдачи IFMP тип 0 потока имеет поле "тип потока"= 0 и поле "длина идентификатора потока"=0; тип 1 потока имеет поле "тип потока"=1 и поле "длина идентификатора потока"=4; тип 2 потока имеет поле "тип потока"=2 и поле "длина идентификатора потока"=3.

Общая обработка сообщений перерасдачи узлами передатчика и равного по положению в линии связи будет детально описана выше. Кроме того, другие особенности элементов сообщения REDIRECT включают управление метками и контроль ошибок. Если метка в поле 428 метки элемента 420 сообщения REDIRECT находится вне диапазона, который может обрабатываться в соответствующей линии связи, то сообщение LABEL_RANGE (диапазон метки) может быть направлено к передатчику элемента сообщения REDIRECT. Это сообщение информирует передатчик о диапазоне меток, который может быть использован при принятии элемент сообщения REDIRECT, определяющий поток как уже перерасданный, то системный узел проигнорирует поле метки в принятом сообщении REDIRECT в сравнении с меткой, записанной для перерасданного потока. Если метки совпадают, то системный узел сбрасывает время жизни перерасданного потока на значение, содержащееся в поле "время жизни" 426 принятого элемента сообщения REDIRECT. Если метки не совпадают, то системный узел игнорирует принятый элемент сообщения, и поток возвращающийся в состояние, установленное по умолчанию. Если системный узел обнаруживает какую-либо ошибку в любом из полей элемента сообщения REDIRECT, то данный конкретный элемент сообщения REDIRECT отбрасывается. Однако любые другие не содержащиеся ошибки элементы сообщения REDIRECT, которые могут быть в том же самом теле сообщения перерасдачи протокола IFMP, не отбрасываются и не подвергают изменением каким-либо образом.

Системный узел отвечает сообщением ERROR соседнему узлу, который передал элемент сообщения REDIRECT, если для данного системного узла на портна версия протокола IFMP в принятом сообщении протокола IFMP. Таким образом, если для системного узла не понятен тип потока в каком-либо из элементов сообщения REDIRECT в принятом сообщении протокола IFMP, то системный узел передает сообщение ERROR для каждого типа потока, который ему

указанный конкретный элемент сообщения REDIRECT.

Как описано выше, сообщение протокола перерасдачи IFMP может быть сообщением RECLAIM (восстановление), которое используется для указания соседнему узлу на необходимость освобождения одного или нескольких потоков от связанных с ними меток, с которыми они в данный момент связаны, и освободить метки для повторного их использования. Системный узел, принимающий элемент сообщения RECLAIM от узла в находящейся линии, освобождает метку и передает узлу в находящейся линии элемент сообщения RECLAIM ACK (подтверждение восстановления) в качестве формального подтверждения приема сообщения RECLAIM. Каждый элемент сообщения RECLAIM в теле сообщения 394 имеет структуру, показанную на фиг.9с. Элемент 432 сообщения RECLAIM (в порядке от старшего бита к младшему) содержит следующие поля: 8-битовое поле 434 типа потока, 8-битовое поле 436 длины идентификатора потока и 16-битовое поле 438 в составе первого 32-битового слова; 32-битовое поле 440 метки в качестве второго 32-битового слова, и идентификатор потока 442, представляющий собой поле в целое кратное 32-битового слова. Поле 434 типа потока определяет тип потока идентификатора потока, содержащегося в поле 442 идентификатора потока, в поле 436 длины идентификатора потока, в поле 440 метки идентификатора потока в целых кратных 32-битового слова. В рассматриваемом варианте осуществления размерное поле 438 не используется и установлено на нуль

системным узлом, передающим элемент сообщения RECLAIM, и игнорируется системным узлом, принимающим элемент сообщения RECLAIM. Поле 440 метки содержит метку, которую следует освободить. Поле 442 идентификатора потока, относительно которого определена метка, указанная в поле 440 метки, должна быть освобождена. Каждый элемент сообщения RECLAIM применяется к одному потоку и одной метке. После того как системный узел примет элемент сообщения RECLAIM, удалит из потока метку, возвратит поток в состояние пересылки, установленное по умолчанию, и освободит метку, системный узел должен сформировать элемент сообщения RECLAIM ACK (подтверждение восстановления).

Элементы сообщения RECLAIM ACK могут группироваться вместе в одно или несколько сообщений RECLAIM ACK и возвращаться к передатчику в качестве подтверждения завершения операции восстановления.

Дополнительными особенностями элемента сообщения RECLAIM являются управление меткой и контроль ошибок. Если системный узел принял элемент сообщения RECLAIM, определяющий неизвестный поток, то системный узел возвращает элемент сообщения RECLAIM ACK с теми же полями метки 440 и идентификатора потока 442 передатчику элемента сообщения RECLAIM. Если системный узел принял элемент сообщения RECLAIM, который указывает на известный поток с меткой в поле 440 метки, элемент сообщения RECLAIM являются управленными меткой и контроль ошибок. Если системный узел принял элемент сообщения RECLAIM, определяющий неизвестный поток, то системный узел возвращает элемент сообщения RECLAIM ACK с теми же полями метки 440 и идентификатора потока 442 передатчику элемента сообщения RECLAIM. Если системный узел принял элемент сообщения RECLAIM, который указывает на известный поток с меткой в поле 440 метки,

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

данном потоке, то системный узел возвращает данный поток от метки и устанавливает поток в состояние пересылки, упоминание по упоминанию, в также формирует элемент сообщения RECLAIM ACK, содержащий действительную метку, с которой поток был ранее связан, передавшему элементу сообщения RECLAIM. Если системный узел обнаруживает ошибку в каком-либо из полей элемента сообщения RECLAIM, данный конкретный элемент сообщения RECLAIM, содержащий ошибку, отбрасывается. Однако любые другие, не содержащие ошибок элементы сообщения RECLAIM, которые могут быть в том же самом теле сообщения восстановления протокола IFMP, не отбрасываются и не подвергаются изменению каким-либо образом. Системный узел отвечает сообщением ERROR создателю узлу, который передал элемент сообщения RECLAIM, если для данного системного узла не понята версия протокола IFMP в принятом сообщении протокола IFMP. Таким образом, если для системного узла не понята версия протокола IFMP, то системный узел передает сообщение ERROR для каждого типа потока, который ему не понятен, соседнему узлу, который передал указанный конкретный элемент сообщения RECLAIM.

Как упомянуто выше, сообщение протокола перераспределения IFMP может представлять собой сообщение RECLAIM ACK, которое используется для подтверждения успешного освобождения метки. После того как системный узел принял исходящий элемент сообщения RECLAIM от узла в сообщении RECLAIM ACK, элемент сообщения RECLAIM ACK передается к узлу, который передал элемент сообщения RECLAIM. Если возможно, то каждый элемент сообщения RECLAIM ACK не следует передавать, пока не будут переданы все данные, находящиеся в очереди для передачи по линии связи, с использованием метки, определенной для освобождения. Каждый элемент сообщения RECLAIM ACK в теле сообщения 394 имеет структуру, показанную на фиг.98. Элемент 444 сообщения RECLAIM ACK (в порядке от старшего бита к младшему) содержит следующие поля: 8-битовое поле 446 типа потока, 8-битовое поле 448 длины идентификатора потока и 16-битовое резервное поле 450 в составе первого 32-битового слова; 32-битовое поле 452 метки в качестве второго 32-битового слова; и идентификатор потока 454, представляющий собой поле в целое кратное 32-битового слова. Поле 446 типа потока определяет тип потока, идентификатор потока, содержащегося в поле 454, идентификатора потока, в поле 448 длины идентификатора потока, определяет длину поля 454.

Идентификатор потока в целых кратных 32-битового слова. В рассматриваемом варианте осуществления резервное поле 450 не используется и установлено на ноль системным узлом, передающим элемент сообщения RECLAIM ACK, и игнорируется системным узлом, принимающим элемент сообщения RECLAIM ACK. Поле 452 метки

определенного потока 454 идентификатора потока. Поле 454 идентификатора потока содержит идентификатор потока из элемента сообщения RECLAIM, в котором было запрошено освобождение метки, указанной в поле 452 метки.

Дополнительными особенностями элемента сообщения RECLAIM являются управление меткой и контроль ошибок. Если системный узел принял элемент сообщения RECLAIM ACK, определяющий поток, для которого не выдавалось сообщение RECLAIM, то элемент сообщения RECLAIM ACK игнорируется. Если системный узел принял элемент сообщения RECLAIM ACK, который указывает на другую метку, отличающуюся от метки, переданной в сообщении RECLAIM для данного потока, то системный узел обрабатывает принятый элемент сообщения RECLAIM ACK, как если бы операция восстановления для метки, переданной в сообщении RECLAIM, была успешной. Если системный узел обнаруживает ошибку в каком-либо из полей элемента сообщения RECLAIM ACK, данный конкретный элемент сообщения RECLAIM ACK, содержащий ошибку, отбрасывается. Однако любые другие, не содержащие ошибок элементы сообщения RECLAIM ACK, которые могут быть в том же самом теле сообщения подтверждения восстановления протокола IFMP, не отбрасываются и не подвергаются изменению каким-либо образом. Системный узел отвечает сообщением ERROR создателю узлу, который передал элемент сообщения RECLAIM ACK с ошибкой, если для данного системного узла не понята версия протокола IFMP в принятом сообщении протокола IFMP. Таким образом, если для системного узла не понята версия протокола IFMP, то системный узел передает сообщение ERROR для каждого типа потока, который ему не понятен, соседнему узлу, который передал указанный конкретный элемент сообщения RECLAIM ACK.

Как описано выше, сообщение протокола перераспределения IFMP может представлять собой сообщения LABEL RANGE (диапазон меток), которые используются в ответ на сообщения REDIRECT (перераспределения), если запрошенная метка в одном или нескольких элементах сообщения REDIRECT находится вне диапазона, который может быть обработан системным узлом, принимающим сообщение перераспределения. Сообщение LABEL RANGE информирует передатчик сообщения REDIRECT о диапазоне меток, который может обрабатываться в данной линии связи. В сообщении LABEL RANGE используется один элемент сообщения, диапазон меток. Элемент сообщения LABEL RANGE в теле сообщения 394 имеет структуру, представленную на фиг.99. Элемент сообщения 456 LABEL RANGE содержит поле 458 минимальной метки в виде первого 32-битового слова и поле 460 максимальной метки в виде второго 32-битового слова. Поле 458 минимальной метки и поле 460 максимальной метки соответственно представляют собой минимальное и максимальное значение метки, которая может быть определена в

конкретной линии связи. Только эти значения меток в пределах диапазона от минимальной метки до максимальной метки (включительно) могут быть определены в сообщении протокола перераспределения IFMP в линии связи.

Как описано выше, сообщение протокола перераспределения IFMP может представлять собой сообщения ERROR (ошибка), которое может передаваться в ответ на любое сообщение протокола перераспределения IFMP. В сообщении ERROR используется один элемент сообщения ошибки. Элемент сообщения ERROR в теле сообщения 394 имеет структуру, представленную на фиг.99. Элемент 462 сообщения ERROR (в порядке от старшего бита к младшему) содержит 8-битовое поле 464 кода ошибки и 24-битовое поле 466 параметров в виде 32-битового слова. Поле 464 кода ошибки определяет, какой тип ошибки имел место. Сообщение ERROR может определять один параметр. Если системный узел обнаруживает ошибку в любом из полей в элементе сообщения протокола перераспределения IFMP, то данный конкретный элемент сообщения с ошибкой отбрасывается и выдается сообщение ERROR. Если системный узел не может осуществить обработку или ему не понята конкретная версия протокола IFMP в принятом сообщении протокола IFMP, то системный узел передает сообщение ERROR с кодом кода ошибки 464, установленным в 1, и с полем 466 параметров, обозначившим самую последнюю версию протокола IFMP, которую передатчик может понять или осуществить соответствующую обработку. Кроме того, если системному узлу не понятен тип потока в каком-либо из элементов сообщения протокола перераспределения IFMP, который вызвал ошибку, то системный узел передает сообщение ERROR с полем 464 кода ошибки, установленным на 2, и с полем 466 параметров, указывающим тип потока, который вызвал ошибку.

2. Передача маркированного потока по линиям передачи ATM данных. Настоящее изобретение использует линии передачи ATM данных для передачи IP-пакетов между системными узлами. Пакеты, передаваемые по линиям передачи ATM данных, представляют собой потоки, маркированные и инкапсулированные в различных образах, в зависимости от типа потока, как указано выше. С использованием классификации потоков настоящих изобретения обеспечивается осуществление обработки разных типов пакетов различных образов (маршрутизация на уровне 2 или коммутация на уровне 3), в зависимости от типа пакета. Кроме того, каждый тип пакета также определяет инкапсуляцию, которая должна использоваться после перераспределения данного типа пакета. В рассматриваемом варианте осуществления изобретения система использует инкапсуляцию для линий передач ATM данных, как описано подробно ниже. Разумеется, инкапсуляция для каждого типа потока могут быть определены для различных технологических линий передачи данных, для различных аппаратных средств коммутации, которые могут использоваться в соответствии с настоящим изобретением. Конкретный поток пакетов может быть

режима передачи (ATM меткой). На фиг.10а показан формат 32-битового поля для ATM метки в рассматриваемой системе. Как описано выше, метка представляет собой идентификатор виртуального маршрута и идентификатор виртуального канала (VPI/VCI), причем предоставляются однонаправленные виртуальные каналы. В порядке от старшего бита к младшему поле 470 ATM метки, показанное на фиг.10а, содержит 4-битовое резервное поле 472, 12-битовое поле 474 VPI и 16-битовое поле 476 VCI. В рассматриваемом варианте осуществления резервное поле 472 установлено на ноль передающим системным узлом и игнорируется системным узлом, принимающим ATM метку. Для линии передачи данных, которая не поддерживает полных 12-битовых VPI, используются старшие биты поля 474 VPI, представляющие собой биты в поле 474 VPI, установленные на ноль. Аналогично для линии передачи данных, которая не поддерживает полных 16-битовых VCI, неиспользованные биты в поле 476 VCI представляют собой старшие биты поля 476, установленные на ноль.

Для любых пакетов в потоке, который не перераспределен, системный узел использует инкапсуляцию IP-пакетов, установленную по умолчанию. Если системный узел принимает решение, что конкретный тип пакета должен быть перераспределен, то системный узел использует инкапсуляцию, выбранную конкретно для данного типа потока. После приема потока системный узел изменит инкапсуляцию, используемую для перераспределения потока, с установленной по умолчанию на иную. Вместо использования для IP-пакетов инкапсуляции, установленной по умолчанию в канале передачи, установленном по умолчанию, системный узел использует другой тип инкапсуляции в зависимости от типа потока, который перераспределен. Следует иметь в виду, что инкапсулированный IP-пакет, который может представлять собой IP-пакет, который сам инкапсулирует сообщение протокола IFMP, передаваемое к (или от) главного компьютера/серверу/рабочей станции, на которых выполняется комплект системного программного обеспечения, базовому коммутационному блоку или шлюзовому блоку коммутации.

Как описано выше, в возможном варианте осуществления изобретения определены три типа потоков: тип 0 потока, тип 1 потока, тип 2 потока. Тип 0 потока используется для изменения инкапсуляции IP-пакетов с установленной по умолчанию. Тип 1 потока используется для пакетов, переносящих данные между прикладными программами, исполняемыми на станциях. Тип 2 потока используется для пакетов, переносящих данные между станциями, без идентификации прикладных задач, которые могут выполняться на станциях.

В настоящем изобретении инкапсуляция, устанавливаемая по умолчанию, для IP-пакетов в линии передачи ATM данных представляет собой инкапсуляцию, описываемую процедурой. Логическое управление линией связи/точка субканала подсоединения (LLCSNAP), как показано на фиг.10b, описывает IP-пакет,

основном инкапсуляция по умолчанию предусматривает снабжение IP-пакета заголовком LLC/SNAP, причем IP-пакет инкапсулируется в составе полезной нагрузки блока данных протокола Ethernet с помощью адаптации типа 5 (AAL-5 CPCS-PDU). Инкапсулированный по умолчанию IP-пакет 480 содержит заголовок LLC/SNAP (24-битовое поле 482 LLC, за которым следует 8-битовая часть заголовка 484 SNAP, в первом 32-битовом слове, и остается 32-битовая часть заголовка 484 SNAP). IP-пакет 488 (который имеет длину целого кратного 32-битового слова), поле 488 заполнителя и конечное поле 490 AAL-5 CPCS-PDU. Поле заполнителя 488 может иметь длину от 0 до 47 байтов, поле 490 равно 8 байтам (4 32-битовых слова). Блок передачи IP-пакета 488, использующего инкапсуляцию по умолчанию, равен 1500 байтов. Пакеты, использующие инкапсуляцию по умолчанию, передаются к передатчику по определенному VRVLC (VR = 0, VCI = 15), который представляет собой установленные по умолчанию идентификаторы виртуального маршрута и виртуального канала для соответствующей аппаратуры (т.е. пакеты передаются по виртуальному каналу, установленному по умолчанию).

Тип 0 пакета используется для изменения инкапсуляции IP-пакетов с установленной по умолчанию. На фиг.10с показан инкапсулированный IP-пакет 492 типа 0. IP-пакет, использующий тип 0 пакета, инкапсулируется непосредственно в полезную нагрузку AAL-5 CPCS-PDU, без присылания заголовка LLC/SNAP. Инкапсулированный IP-пакет 492 типа 0 включает в себя IP-пакет 494 (имеющий длину, равную целому кратному 32-битового слова), поле заполнителя 498 и поле хвостовика 498 AAL-5 CPCS-PDU. Поле заполнителя 498 может занимать от 0 до 47 байтов, а поле хвостовика 498 равно 8 байтам (четыре 32-битовых слова). Блок передачи IP-пакета 494, использующего инкапсуляцию по умолчанию, равен 1500 байтов. Пакеты, принадлежащие к потоку, передаваемому из установленного канала, передаются по виртуальному каналу, по умолчанию инкапсуляция типа 0 пакета и передаются в поле метки VRVLCI, определенные в поле метки элемента сообщения REDIRECT (переадресация) протокола IFMP. Инкапсулированный в IP-пакете 494 (элемент сообщения REDIRECT, инкапсулированный в IP-пакете 494, передается с инкапсуляцией типа 0 пакета).

Инкапсуляция, установленная по умолчанию и связанная с типом 0 пакета, не предусматривает удаленно как-либо поля из инкапсулированного IP-пакета. Однако инкапсуляция в соответствии с типом 1 пакета и типом 2 пакета связаны с удалением определенных полей из IP-пакета. Если эти поля удаляются, то остаются узел, который выдал сообщение REDIRECT, запоминает удаленные поля и связывает эти поля с виртуальным АТМ маналом, определенным в АТМ метке. Соответственно, полный IP-пакет может быть восстановлен в месте назначения с использованием входящей АТМ метки для доступа к запомненным полям.

переносимых данных между прикладными задачами, выполняемыми на станциях. На фиг.10d, иллюстрируется IP-пакет инкапсулированный в соответствии с типом 1 пакета. IP-пакеты, использующие инкапсуляцию типа 1 пакета, по существу декомпируются и выбранные части инкапсулируются непосредственно в полезную нагрузку AAL-5 CPCS-PDU, без присылания заголовка LLC/SNAP. Инкапсулированный согласно типу 1 пакета IP-пакет 500 содержит 8-битовое поле 502 полной длины и 16-битовое поле 504 идентификации из IP-заголовка декомпированного IP-пакета в качестве первого 32-битового слова. Значение поля 502 полной длины не изменяется, а сохраняется инкапсулированный согласно типу 1 пакета IP-пакет 500 также содержит 8-битовое поле 506 флага, 12-битовое поле 508 смещения фрагмента и 16-битовое поле 510 контрольной суммы из IP-заголовка декомпированного IP-пакета в качестве второго 32-битового слова. Передаваемое значение поля 510 контрольной суммы представляет собой значение контрольной суммы, которое было бы вычислено для полного IP-заголовка, если бы поле TTL было установлено в ноль. Поля "версия", "IHL", "TOS", "TTL", "протокол", "адрес источника", "адрес места назначения" в IP-заголовке не передаются в виде части инкапсулированного согласно типу 2 пакета IP-пакета 520. В отличие от инкапсуляции согласно типу 1, поле, первая четыре байта, непосредственно следующие на IP-заголовке, передаются в виде части инкапсулированного согласно типу 2 пакета IP-пакета 520. Кроме того, IP-пакет 520, инкапсулированный в соответствии с типом 2 пакета, содержит поле данных 532. За полем данных 532 следует поле заполнителя 534 и поле хвостовика 536 AAL-5 CPCS-PDU. Поле заполнителя 534 может иметь длину от 0 до 47 байтов, а поле хвостовика 536 равно 8 байтам (четыре 32-битовых слова). Передаваемый блок IP-пакета, использующего инкапсуляцию согласно типу 2 пакета, содержит 1484 байта. Пакеты, принадлежащие к потоку, передаваемому с "событием", за исключением класса сообщений "события", остальные четыре класса представляют собой классы сообщений типа "запрос-ответ", причем каждый имеет формат для сообщения запроса и формат для ответа при успешной операции. Если только не указано иное, сообщение, если оно не успешной операции является тем же самым, что и сообщение запроса, которое обусловило этот успех, за исключением того, что поле "код" указывает характер такого неуспеха операции. Помимо четырех классов сообщений типа "запрос-ответ", протокол GSNMP включает класс сообщений "события", который позволяет коммутатору генерировать всевозможные сообщения "события" для информирования контроллера коммутации об всевозможных событиях. Поскольку сообщения "события" не подтверждаются при приеме контроллером коммутации, сообщения "события" имеют один формат. В рассматриваемом варианте осуществления имеется множество различных типов сообщений, т.е. функций сообщений протокола GSNMP. Каждый из пяти классов сообщений протокола GSNMP, за исключением "управления портом", имеет ряд различных типов сообщений.

инкапсулируются непосредственно в полезную нагрузку AAL-5 CPCS-PDU, без присылания заголовка LLC/SNAP. Инкапсулированный согласно типу 2 пакета IP-пакет 520 содержит 16-битовое поле 522 полной длины и 16-битовое поле 524 идентификации из IP-заголовка декомпированного IP-пакета в качестве первого 32-битового слова. Значение поля 522 полной длины не изменяется, а сохраняет полную длину IP-пакета до декомпиляции. Инкапсулированный согласно типу 2 пакета IP-пакет 520 также содержит 8-битовое поле 526 флага, 12-битовое поле 528 смещения фрагмента и 16-битовое поле 530 контрольной суммы из IP-заголовка декомпированного IP-пакета в качестве второго 32-битового слова. Передаваемое значение поля 530 контрольной суммы представляет собой значение контрольной суммы, которое было бы вычислено для полного IP-заголовка, если бы поле TTL было установлено в ноль. Поля "версия", "IHL", "TOS", "TTL", "протокол", "адрес источника", "адрес места назначения" в IP-заголовке не передаются в виде части инкапсулированного согласно типу 2 пакета IP-пакета 520. В отличие от инкапсуляции согласно типу 1, поле, первая четыре байта, непосредственно следующие на IP-заголовке, передаются в виде части инкапсулированного согласно типу 2 пакета IP-пакета 520. Кроме того, IP-пакет 520, инкапсулированный в соответствии с типом 2 пакета, содержит поле данных 532. За полем данных 532 следует поле заполнителя 534 и поле хвостовика 536 AAL-5 CPCS-PDU. Поле заполнителя 534 может иметь длину от 0 до 47 байтов, а поле хвостовика 536 равно 8 байтам (четыре 32-битовых слова). Передаваемый блок IP-пакета, использующего инкапсуляцию согласно типу 2 пакета, содержит 1484 байта. Пакеты, принадлежащие к потоку, передаваемому с "событием", за исключением класса сообщений "события", остальные четыре класса представляют собой классы сообщений типа "запрос-ответ", причем каждый имеет формат для сообщения запроса и формат для ответа при успешной операции. Если только не указано иное, сообщение, если оно не успешной операции является тем же самым, что и сообщение запроса, которое обусловило этот успех, за исключением того, что поле "код" указывает характер такого неуспеха операции. Помимо четырех классов сообщений типа "запрос-ответ", протокол GSNMP включает класс сообщений "события", который позволяет коммутатору генерировать всевозможные сообщения "события" для информирования контроллера коммутации об всевозможных событиях. Поскольку сообщения "события" не подтверждаются при приеме контроллером коммутации, сообщения "события" имеют один формат. В рассматриваемом варианте осуществления имеется множество различных типов сообщений, т.е. функций сообщений протокола GSNMP. Каждый из пяти классов сообщений протокола GSNMP, за исключением "управления портом", имеет ряд различных типов сообщений.

Для инкапсуляции согласно типу 0 пакета, типу 1 пакета и типу 2 пакета системный узел, принимающий сообщение, передающийся, передающее узлом входящей линии, запоминает удаленные поля и связывает эти поля с виртуальным каналом АТМ путем АТМ метки, обеспечивая ксперированную информацию доступа для передаваемых пакетов, как описано выше.

В. Протокол GSNMP
Системное программное обеспечение также использует протокол GSNMP для установления связи по АТМ линии передачи данных между контроллером коммутации и аппаратными средствами АТМ коммутации базового коммутационного блока системы, тем самым обеспечивая коммутацию на маршрутизацию согласно протоколу IP на уровне 3, если это необходимо. В частности, протокол GSNMP представляет собой универсальный асимметричный протокол для

в виртуальном канале, установленном при инициализации в линии передачи АТМ данных между контроллером коммутации и АТМ коммутатором. Один контроллер коммутации может использовать множество отдельных виртуальных каналов для управления множеством АТМ коммутаторов. Протокол GSNMP также включает в себя протокол близости GSNMP, который используется для синхронизации состояний в линии передачи АТМ данных между контроллером коммутации и АТМ коммутатором, для определения идентификации узла на другом конце линии связи и для определения изменения в идентификации этого узла.

Протокол GSNMP позволяет контроллеру коммутации устанавливать и отменять соединения через АТМ коммутатор, добавляя и исключая элементы в соединении от одной точки к множеству точек, управлять портами коммутатора, запрашивать информацию конфигурирования и статистические данные. Протокол GSNMP также позволяет АТМ коммутатору информировать контроллер коммутации о таких событиях, как отказ в линии связи.

Как упомянуто выше, протокол GSNMP представляет собой протокол типа "вздуши-поднимаешь". Контроллер коммутации выдает сообщение запроса на коммутатор. Каждое сообщение запроса показывает, требуется ли ответ от коммутатора, и содержит идентификатор операции для обеспечения того, чтобы ответ был связан с конкретным запросом. Коммутатор отвечает ответными сообщениями, указывающими успех или неуспех операции. В рассматриваемом варианте осуществления протокол GSNMP предусматривает пять классов сообщений. Управление соединением, "Управление портом", "Статистика", "конфигурирование", "события". За исключением класса сообщений "события", остальные четыре класса представляют собой классы сообщений типа "запрос-ответ", причем каждый имеет формат для сообщения запроса и формат для ответа при успешной операции. Если только не указано иное, сообщение, если оно не успешной операции является тем же самым, что и сообщение запроса, которое обусловило этот успех, за исключением того, что поле "код" указывает характер такого неуспеха операции. Помимо четырех классов сообщений типа "запрос-ответ", протокол GSNMP включает класс сообщений "события", который позволяет коммутатору генерировать всевозможные сообщения "события" для информирования контроллера коммутации об всевозможных событиях. Поскольку сообщения "события" не подтверждаются при приеме контроллером коммутации, сообщения "события" имеют один формат. В рассматриваемом варианте осуществления имеется множество различных типов сообщений, т.е. функций сообщений протокола GSNMP. Каждый из пяти классов сообщений протокола GSNMP, за исключением "управления портом", имеет ряд различных типов сообщений.

Протокол GSNMP включает сообщения протокола близости GSNMP, которому

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

Протокол близости GSMR используется для установления синхронизации в линии передачи АТМ данных и поддерживает каналовете. За исключением сообщений протокола близости GSMR никакие другие сообщения GSMR не могут передаваться по линии передачи АТМ данных до тех пор, пока протокол близости GSMR не установит синхронизацию в состоянии. Все сообщения GSMR, принятые по линии передачи АТМ данных, которые в текущий момент не достигли синхронизации в состоянии, отбрасываются.

В настоящем изобретении пакеты протокола GSMR имеют переменную длину и инкапсулированы непосредственно в ААЛ-5 CPDUS-PDU с приписанным заголовком LLC/SNAP, аналогичным инкапсуляции по умолчанию IP-пакетов в линиях передачи АТМ данных, описанных выше со ссылкой на фиг.10b. На фиг.11a показан инкапсулированный пакет 540 протокола GSMR. В основном инкапсуляция по умолчанию предусматривает приписывание заголовка LLC/SNAP к пакету протокола GSMR, который инкапсулирован в составе полезной нагрузки ААЛ-5 CPDUS-PDU. Инкапсулированный по умолчанию GSMR-пакет 540 содержит заголовок LLC/SNAP (24-битовое поле 542 LLC, за которым следует 8-битовая часть заголовка 544 SNAP в первом 32-битовом слове, и оставшаяся 32-битовая часть заголовка 544 SNAP) GSMR сообщения 548 (которое имеет длину целого кратного 32-битового слова), поле 546 заголовка и последнее поле 550 ААЛ-5 CPDUS-PDU. Поле заголовка 546 может иметь длину от 0 до 47 байтов, поле 550 ААЛ-5 CPDUS-PDU равно 8 байтам (четыре 32-битовых слова). Передаваемый блок GSMR сообщения 546, используемый инкапсуляцией по умолчанию, равен 1500 байтов. Пакеты, использующие инкапсуляцию по умолчанию, передаются к установленному по умолчанию устройству, т.е. к установленному по умолчанию виртуальному каналу.

На фиг.11b показана структура сообщения 522 протокола близости GSMR, которое может содержаться в поле 546 сообщения протокола GSMR инкапсулированного пакета 540 протокола GSMR по фиг.11a. Как показано на фиг.11b, сообщение 522 протокола близости GSMR содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле "версия" 554, 6-битовое поле "тип сообщения" 556, 8-битовое поле "результат" 558 и 8-битовое поле "код" 560. В качестве первого 32-битового слова; поле 562 "экземпляр передатчика" в качестве второго 32-битового слова; поле 564 "порт передатчика" в качестве третьего 32-битового слова; поле 566 "имя приемника" в виде следующих 48 битов; поле 568 "имя приемника" в виде следующих 48 битов; поле 570 "порт приемника" в виде следующих 32 битов; и поле 572 "экземпляр приемника" в виде следующих 32 битов. При обсуждении сообщений протокола GSMR под понятием "передатчик" понимается узел, который передает сообщение протокола GSMR по линии передачи АТМ данных. Таким образом может быть контроллер коммутации или АТМ коммутатор.

GSMR поле 554 "версия" определяет версию протокола GSMR, которая используется в данный момент времени. Поле 554 "тип сообщения" устанавливается на конкретное значение (тип сообщения = 96) для определения сообщения протокола близости GSMR. Поле 556 "результат", на качество сообщения протокола близости GSMR, устанавливается в нуль при успешном узлом и игнорируется узлом, принявшим сообщение протокола близости GSMR.

Поле 560 кода для сообщений протокола близости GSMR определяет функцию сообщения. В раскрываемом варианте осуществления имеются четыре возможных значения поля 560 "Code", т.е. функции сообщения протокола близости GSMR: SYN (сообщение синхронизации, Code= 0), SYNACK (сообщение подтверждения синхронизации, Code=1), RSTACK (сообщение подтверждения сброса, Code=2), ACK (сообщение подтверждения, Code=3). В каждом системном узле необходим таймер для периодического генерирования сообщений SYN, SYNACK, ACK протокола GSMR. Для целей протокола близости GSMR конкретный узел имеет три возможных состояния для конкретной линии связи: SYNSENT (сообщение синхронизации передано), SYNRCVD (сообщение синхронизации принято), ESTAB (синхронизация установлена). Синхронизация достигается в линии связи (когда узел достигает состояния ESTAB для данной линии) необходима, прежде чем узлы смогут передавать сообщения протокола GSMR, которые не являются сообщениями протокола близости GSMR. В раскрываемом варианте осуществления период таймера равен 1 секунде, но могут быть определены и другие длительности периода таймера. Если значение таймера истекло, и системный узел находится в состоянии SYNSENT, то системный узел сбрасывает таймер и передает сообщение SYN протокола близости GSMR. Если значение таймера истекло, и системный узел находится в состоянии SYNRCVD, то системный узел сбрасывает таймер и передает сообщение SYNACK протокола близости GSMR. Если значение таймера истекло и системный узел находится в состоянии ESTAB, то системный узел сбрасывает таймер и передает сообщение ACK протокола близости GSMR.

В сообщениях SYN, SYNACK, ACK протокола близости GSMR, ACK, например, передатчика 562 представляет собой номер "экземпляра" передающего узла линии связи. Указывая конкретный экземпляр линии, номер экземпляра представляет 32-битовое ненулевое число, для которого гарантировано его уникальность в пределах ближайшего прошлого и его изменения, если линия или системный узел вводится в действие после отказа или если идентификация узла на другом конце линии связи изменится. Соответственно каждая линия имеет свой собственный уникальный номер экземпляра. Экземпляр передатчика 562 используется для определения того, когда линия вновь стала работоспособной после отказа, или когда идентификация узла на другом конце линии

сообщения RSTACK протокола близости GSMR, например, передатчика 562 устанавливается на значение поля 572 экземпляра приемника из входящего сообщения протокола близости GSMR, которое вызвало генерирование сообщения RSTACK.

В сообщениях SYN, SYNACK, ACK протокола близости GSMR поле 564 порта передатчика представляет собой локальный номер порта линии связи, по которой передается сообщение протокола GSMR. Как указано выше, номера портов представляют собой локально присваиваемые 32-битовые значения. В сообщении RSTACK протокола близости GSMR поле 564 порта передатчика устанавливается на значение поля 570 приемника из входящего сообщения протокола близости GSMR, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости GSMR поле 566 имени передатчика представляет собой имя передающего узла. 48-битовое поле 566 имени передатчика является уникальным в операционном контексте базового коммутационного блока. Например, для поля "имя передатчика" может быть использован адрес IEEE 802 MAC. Для сообщения RSTACK протокола близости GSMR поле 566 имени передатчика устанавливается на значение поля 566 имени приемника из входящего сообщения протокола близости GSMR, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости GSMR поле 568 имени приемника представляет собой имя узла, который, как представляется, передается сообщению узлу, находящимся на другом конце линии передачи АТМ данных. Если в передающем узле не известно имя этого узла, то поле 568 имени приемника устанавливается на нуль. Для сообщения RSTACK протокола близости GSMR поле 568 имени приемника устанавливается на значение поля 568 имени передатчика из входящего сообщения протокола близости GSMR, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости GSMR поле 570 порта приемника представляет собой то, что, как представляется, передается узлу, является локальным номером порта, выделенным для линии узлом на другом конце линии передачи АТМ данных. Если в передающем узле не известен номер порта этого узла, то поле 570 порта приемника устанавливается на нуль. Для сообщения RSTACK протокола близости GSMR поле 570 порта приемника устанавливается на значение поля 564 порта передатчика из входящего сообщения протокола близости GSMR, которое вызвало генерирование сообщения RSTACK.

Для сообщений SYN, SYNACK, ACK протокола близости GSMR, поле 572 экземпляра приемника представляет собой то, что, как представляется, передается сообщению узлу, является текущим номером экземпляра, выделенным для линии узлом на другом конце линии передачи. Если в передающем узле не известно текущий номер экземпляра на другом конце линии, то поле 572 экземпляра приемника устанавливается на

близости GSMR, поле 572 экземпляра приемника устанавливается на значение поля 562 экземпляра передатчика из входящего сообщения протокола близости GSMR, которое вызвало генерирование сообщения RSTACK.

На фиг.11c представлена диаграмма, иллюстрирующая работу передающего узла после приема входящего сообщения протокола близости GSMR. После запуска системы передающий узел принимает пакет протокола близости GSMR (отл 562). На этапе 584 передающий узел определяет, является ли входящее сообщение протокола близости GSMR сообщением RSTACK. Если входящее сообщение протокола близости GSMR является сообщением RSTACK (например, сообщение SYN, SYNACK, ACK), то передающий узел работает в соответствии с тем, как показано на диаграмме состояний на фиг.11d. Если входящее сообщение протокола близости GSMR является сообщением RSTACK, то передающий узел проверяет на этапе 5844, совпадают ли значения полей экземпляра передатчика, порта передатчика и имени передатчика во входящем сообщении RSTACK со значениями, запомненными из предыдущего сообщения посредством операции обновления верификатора равного по положению. Для протокола близости GSMR операция обновления верификатора равного по положению определяется как запоминание значений полей экземпляра передатчика, порта передатчика и имени передатчика из сообщения SYN или SYNACK, принятого от сообщения SYN или SYNACK, принятого от узла на другом конце линии. Если на этапе 584 установлено, что указанные значения совпадают, то передающий узел на этапе 586 определяет, совпадают ли значения полей экземпляра приемника, порта приемника и имени приемника во входящем сообщении RSTACK со значениями экземпляра приемника, порта приемника и имени приемника, используемыми для всех сообщений SYN, SYNACK или ACK, передаваемых с порта, на который было принято входящее сообщение RSTACK. Если на этапе 586 установлено, что указанные значения совпадают, то передающий узел на этапе 588 определяет, находится ли передающий узел в состоянии SYNSENT. Если передающий узел не находится в состоянии SYNSENT, то передающий узел осуществляет установку линии в исходное состояние на этапе 590. Если на этапе 584 или на этапе 586 установлено, что значения не совпадают, или передающий узел не находится в состоянии SYNSENT, то передающий узел на этапе 592 отбрасывает входящее сообщение RSTACK и ожидает прихода другого пакета. Соответственно, когда в передающий узел приходит сообщение RSTACK протокола близости GSMR, передающий узел устанавливает линию в исходное состояние, как показано на этапе 594, 596, 598, 600. На этапе 594 передающий узел генерирует новый номер экземпляра для линии. Затем передающий узел на этапе 596 отмечает (т.е. устанавливает в нуль) запомненные значения поля экземпляра передатчика, порта передатчика и имени передатчика, которые были запомнены в процессе операции

RU 2189072 C2

RU 2189072 C2

положению. На этапе 598 передающий узел передает сообщение SYN протокола близости GSMР и на этапе 600 входит в состояние SYNSENT. Передающий узел затем принимает другой пакет для его обработки.

На фиг.11д представлена диаграмма состояния, иллюстрирующая работу передающего узла, когда входящее сообщение протокола RSTACK для GSMР не является следующим состоянием. "4Б" определяется следующим образом: поля эсэмплера передающего, порта передающего и имени передающего во входящем сообщении совпадают со значениями, запомненными из предыдущего сообщения посредством операции обновления верификатора равного по положению. Условие "4С" на фиг.11д определяет следующие образцы: эсэмплер приемника, порт приемника и имя приемника во входящем сообщении совпадают со значениями эсэмплера передающего, порта передающего и имени передающего, используемыми в настоящее время для исходящих сообщений SYN, SYNACK, ACK. На фиг. 11д условие "А" означает, что передающий узел принимает входящее сообщение SYNACK протокола близости GSMР и что условие "4С" удовлетворено. Условие "В" означает, что передающий узел принимает входящее сообщение SYNACK протокола близости GSMР и что условия "4В" и "4С" не удовлетворены.

Если передающий узел находится в состоянии 602 SYNSENT и принимает входящее сообщение SYN протокола близости GSMР от равного по положению на другом конце линии связи, то передающий узел выполняет операцию обновления верификатора равного по положению и передает сообщение SYNACK протокола близости GSMР к равному по положению (показано как этап 604). Затем передающий переходит из состояния 602 SYNSENT в состояние 606 SYNRCVD. Если передающий принимает входящее сообщение SYN протокола близости GSMР, находясь в состоянии 606 SYNRCVD, то на этапе 604 передающий выполняет операцию обновления верификатора равного по положению и передает сообщение SYNACK протокола близости GSMР к равному по положению, но остается в состоянии 606 SYNRCVD. Если передающий находится в состоянии 606 SYNRCVD и либо состояние В, либо состояние С удовлетворено, то передающий передает сообщение RSTACK протокола близости GSMР к равному по положению (показано как этап 608) и остается в состоянии 606 SYNRCVD. Если передающий находится в состоянии 606 SYNRCVD и состояние С удовлетворено, то передающий передает сообщение ACK протокола близости GSMР к равному по положению (показано как этап 610) и переходит в состояние 612

состоянии 608 SYNRCVD и состояние А удовлетворено, то передачик выполняет операцию идентификатора обновления верификатора равного по положению, передает сообщение ACK протокола близости GSMР к равному по положению (показано как этап 614) и переходит в состояние 612 ESTAB. Передачик продолжает находиться в состоянии 612 ESTAB, если передачик принимает сообщение SYN или SYNACK протокола близости GSMР, или если удовлетворено условие С. Если условие D удовлетворено, когда передачик находится в состоянии 612 ESTAB, то передачик остается в состоянии 612 ESTAB и передает сообщение RSTACK протокола близости GSMР (показано как этап 608). В состоянии 602 SYNSENT, если либо передачик принимает сообщение ACK протокола близости GSMР, либо удовлетворено условие В, то передачик остается в состоянии 602 SYNSENT и передает сообщение RSTACK протокола близости GSMР (этап 608). Если находится в состоянии 602 SYNSENT, то передачик выполняет операцию обновления верификатора равного по положению и передает сообщение ACK протокола близости GSMР (этап 614) и переходит в состояние 612 ESTAB.

Помимо сообщений протокола близости GSMР другие типы сообщений 546 протокола GSMР включают сообщения управления соединением (CM), которые представляют собой сообщения типа "запрос-ответ". В базовом коммутиционном блоке контроллер коммутации использует сообщения управления соединением протокола GSMР для установления, отмены, изменения и проверки соединений виртуального канала через АТМ коммутатор. Сообщения управления соединением протокола GSMР могут выдаваться независимо от статуса порта коммутации, и соединения могут устанавливаться или отменяться, когда порт коммутатора задействован, или иным образом недоступен. Сообщения управления соединением включают следующие: "добавить переход", "удалить переход", "удалить ветвь", "переместить корень", "переместить переход". Как отмечено выше, соединение виртуального канала является однонаправленным и включает в себя входной виртуальный канал и, по меньшей мере, один выходной виртуальный канал или переход ("ветвь"). Это означает, что однонаправленное соединение имеет одну выходную ветвь, а многонаправленное соединение имеет две или более выходных ветвей.

Сообщение "добавить переход" представляет собой сообщение управления соединением протокола GSMР, используемое для установления соединения виртуального канала или для добавления дополнительной ветви к имеющемуся соединению виртуального канала. В рассмотряемом варианте осуществления изобретения не делается различий между однонаправленными и многонаправленными соединениями. Первое сообщение "добавить переход" для конкретного входного порта, входного

выходного идентификатора виртуального канала устанавливает однонаправленное соединение. Второе сообщение "добавить переход" с теми же самыми значениями входного порта, входного идентификатора виртуального маршрута и входного идентификатора виртуального канала преобразует однонаправленное соединение в многонаправленное соединение путем добавления второй выходной ветви. Последующие выходные ветви могут добавляться аналогичным образом с помощью последующих сообщений "добавить переход". Кроме того, сообщения "добавить переход" может использоваться для проверки состояния соединения, записанного в АТМ коммутатора. Сообщение "удалить переход" представляет собой сообщение управления соединением протокола GSMР, используемое для удаления соединения виртуального канала. Например, использование сообщения "удалить переход" для многонаправленного соединения виртуального канала с двумя ветвями удаляет одну ветвь, преобразуя многонаправленное соединение в однонаправленное. Сообщения "удалить переход" может также использоваться для удаления соединения путем удаления последней ветви в соединении виртуального канала. Еще одно сообщение протокола GSMР сообщение "удалить ветвь" используется для удаления целого виртуального соединения путем удаления всех оставшихся ветвей соединения. Сообщения "проверить дерево" представляют собой сообщения протокола GSMР, используемое для проверки числа ветвей в соединении виртуального канала. Сообщение "удалить все" представляет собой сообщение протокола GSMР, используемое для удаления всех соединений для входного порта коммутатора. Сообщение "переместить корень" представляет собой сообщение протокола GSMР, используемое для перемещения всего дерева виртуальных соединений от текущего входного порта, входного идентификатора виртуального маршрута и входного идентификатора виртуального канала к новым значениям входного порта, входного идентификатора виртуального маршрута и входного идентификатора виртуального канала. Еще одно сообщение протокола GSMР, сообщение "переместить переход" используется для перемещения одной выходной ветви соединения виртуального канала от его текущих значений выходного порта, выходного идентификатора виртуального маршрута и входного идентификатора виртуального канала к новым значениям выходного порта, выходного идентификатора виртуального маршрута и выходного идентификатора виртуального канала для того же самого соединения виртуального канала.

На фиг. 12 показана структура обобщенного сообщения 620 управления соединением протокола GSMР, используемого в качестве запроса и ответа для сообщений "добавить переход", "удалить ветвь", "удалить все", "переместить корень", "удалить ветвь". Сообщение 620 представляет соединение протокола GSMР, которое содержится в поле 546 сообщения протокола GSMР инкапсулированного пакета

Как показано на фиг.12, обобщенное сообщение 520 управления соединением протокола GSMР содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле 622 "версия", 8-битовое поле 624 "тип сообщения", 8-битовое поле 626 "результат", 8-битовое поле 628 "код", 32-битовое поле 630 "идентификатор операции", 32-битовое поле 632 "номер сеанса порта", 32-битовое поле 634 "выходной порт", 32-битовое поле 636 "входная метка", которое включает 4-битовое слово 638, установленное на ноль, 12-битовое поле 638 "входной VPI", 16-битовое поле 640 "входной VCI", 32-битовое поле 642 "выходной порт", 32-битовое поле 644 "выходная метка", которое включает 8-битовое слово 644, установленное на ноль, 12-битовое поле 646 "выходной VPI", 16-битовое поле 648 "выходной VCI", 8-битовое поле 650 "чисто переходов", 8-битовое резервное поле 652 и 8-битовое поле 654 "приоритет".

За исключением сообщений протокола близости GSMР все сообщения протокола GSMР включают поле 622 "версия", поле 624 "тип сообщения", поле 626 "результат", поле 628 "код", поле 630 "идентификатор операции", которые используются принципиально одним и тем же способом. Например, поле 622 2Версия в сообщении протокола GSMР определяет версию устанавливаемого на конкретное значение для определения типа сообщения протокола GSMР. Например, сообщение "добавить переход" управляет соединением протокола GSMР присваивается конкретное значение для поля 624 "тип сообщения", в других типах сообщений присваиваются другие конкретные значения.

Для сообщения протокола GSMР, которое является сообщением запроса, поле 626 "результат" указывает, требуется ли ответ на сообщение запроса, если результат выполнения является успешным. Поле 626 "результат" в сообщении запроса может содержать значения для NoSuccessAck (указывая, что при успешном результате не требуется ответа) или значения для AckAll (указывая, что при успешном результате требуется ответ). Для некоторых типов сообщений запроса протокола GSMР AckAll устанавливается по умолчанию, а значение NoSuccessAck в поле "результат" 626 игнорируется. Для сообщения протокола GSMР, которое является сообщением ответа, поле 626 "результат" может содержать значения для успешного результата (указывая, что запрос был успешным), или отсутствия успешного результата (указывая, что результат был неудачным). Сообщение успешного ответа протокола GSMР не передается до тех пор, пока запрос не будет успешно завершен. Сообщение успешного ответа протокола GSMР является копией возвращенного сообщения запроса, содержащего успешный результат. Для сообщения запроса протокола GSMР, которое не имеет успешного результата, формируются ответное сообщение, указывающее об этом. Ответное сообщение протокола GSMР, указывающее об этом, представляет собой

RU 2189072 C2

RU 2189072 C2

RU 2189072 C2

RU 2189072 C2

сообщения, возвращенного с полем 626 "результат", указывающим собой коммутатор, выдающий ответное сообщение протокола GSMР, указывающее наличие сбоя, в ответ на неудачный результат сообщения запроса, не модифицирует состояние соединения в коммутаторе.

В сообщении протокола GSMР, которое является сообщением запроса, поле 628 "код" может обеспечивать дополнительную информацию относительно результата. Например, поле 628 "код" может содержать код ошибки, определяющий тип ошибки, вызвавшей сбой. Следует иметь в виду, что множество различных типов кодов могут быть определены для использования в поле 628 "код". Примерами кодов ошибок могут служить следующие: отказ, специфический для конкретного типа сообщения; неопределенная причина; не передаваемая другим типом кода сбоя; недостаточное сообщение запроса; определенное сообщение запроса не выполняется на данном коммутаторе; недостаточный номер севанса порта; по меньшей мере один определенный порт не существует; по меньшей мере один определенный порт не работает; по меньшей мере один определенный VPI/VCI находится вне диапазона для по меньшей мере одного определенного порта; определенное соединение не существует; определенная выходная ветвь не существует; определенная выходная ветвь уже установлена для конкретного многократного соединения на определенном многократном порту; достигнут максимальный предел для мультиматричных поддерживаемых коммутаторов; достигнут максимальный предел для ветвей, который может поддерживаться определенным многократным соединением; общая проблема, связанная с возможностями многократного соединения, поддерживаемого коммутатором. Разумеется, могут быть использованы и другие коды. Кроме того, поле 628 "код" может обеспечивать другую информацию в ответном сообщении успешного результата или сообщения типа "сбой". Поле 628 "код" не используется в сообщениях запроса протокола GSMР и устанавливается в ноль.

Поле 630 "идентификатор операции" используется для того, чтобы указать сообщение запроса протокола GSMР с сообщением ответа протокола GSMР. В сообщении запроса протокола GSMР контроллер коммутации выбирает значение идентификатора операции для поля 630. В сообщении ответа протокола GSMР значение идентификатора операции для поля 630 устанавливается на значение, определенное в сообщении запроса, на которое отвечает коммутатор.

Соответствующее сообщение ответа протокола GSMР. Поскольку сообщение типа "сбой" протокола GSMР не требует ответа, то в нем поле 630 идентификатора операции установлено в ноль. Следует иметь в виду, что приведенное выше описание полей "версия", "тип сообщения", "результат", "код", "идентификатор операции" применимо ко всем сообщениям протокола GSMР, за исключением сообщений протокола GSMР. Отличия от общего описания

Для сообщений управления соединением протокола GSMР поле 632 "номер севанса порта" обеспечивает номер севанса для входного порта. В частности, значение поля 632 "номер севанса порта" указывает номер севанса порта для входного порта коммутатора, указанного в поле 634 "входной порт". Каждый порт коммутатора поддерживает номер севанса порта, присвоенный коммутатором. Номер севанса порта остается неизменным, пока порт находится в работоспособном состоянии. Однако если порт становится работоспособным после отказа, то генерируется новый отключающийся номер севанса порта. Предпочтительно новый номер севанса порта генерируют случайным образом. Если контроллер коммутации передает сообщение запроса управления соединением протокола GSMР, которое имеет недостаточное поле 632 "номер севанса порта", то коммутатор реагирует сообщением запроса управления соединением протокола GSMР путем посылки ответного сообщения управления соединением, указывающего наличие сбоя, в котором поле 628 "код" указывает действительный номер севанса порта, вызвавший сбой. Текущий номер севанса порта может быть получен с использованием сообщения конфигурирования протокола GSMР.

В сообщении управления соединением протокола GSMР поле 634 "входной порт" указывает входной порт коммутатора с использованием 32-битового значения, присвоенного коммутатором. Поле 638 "входной VPI" идентифицирует виртуальный ATM маршрут, поступающий на входной порт коммутатора, указанный полем 634 "входной порт", а поле 640 "входной VCI" идентифицирует виртуальный ATM канал, поступающий по виртуальному маршруту, указанному полем 638 "входной VPI".

В сообщении управления соединением протокола GSMР поле 642 "выходной порт" указывает выходной порт коммутатора с использованием 32-битового значения, присвоенного коммутатором. Поле 646 "выходной VPI" идентифицирует виртуальный ATM маршрут, выходящий из выходного порта коммутатора, указанного полем 642 "выходной порт", а поле 648 "выходной VCI" идентифицирует виртуальный ATM канал, выходящий по виртуальному маршруту, указанному полем 646 "выходной VPI".

Для сообщения управления соединением протокола GSMР поле 650 "число переходов" дает число выходных ветвей в соединении виртуального канала. Поле 650 используется в сообщении "проверить дерево" управления соединением протокола GSMР. Для всех сообщений управления соединением протокола GSMР поле 650 устанавливается на ноль. Передающийся узлом и ингибируется приемным узлом. В размаркированном состоянии резервное поле 652, которое не используется для сообщений управления соединением протокола GSMР, устанавливается на ноль. Передающийся узлом и ингибируется приемным узлом.

Поле 654 "приоритет" в сообщении управления соединением протокола GSMР определяет приоритет соединения. Наивысший приоритет обозначается нулем, а

где q - число приоритетов, которое может поддерживать выходной порт коммутатора. Число q для каждого выходного порта коммутатора может быть получено из сообщения конфигурирования порта GSMР. Каждое соединение виртуального канала может быть установлено с определенным качеством обслуживания, путем присвоения ему приоритета при установлении. Для соединения виртуального канала, которые совместно используют один и тот же выходной порт, ATM элемент данных в соединении с более высоким приоритетом с большей вероятностью будет отправлен из коммутатора, чем ATM элемент данных в соединении с менее высоким приоритетом, если оба они присутствуют в коммутаторе в одно и то же время. Поле 654 "приоритет" используется в сообщениях "добавить переход" и "переместить переход" управления соединением протокола GSMР. Если сообщение запроса управления соединением протокола GSMР ("добавить переход" или "переместить переход") имеет значение в поле 654 "приоритет", которое не поддерживается коммутатором, то коммутатор присваивает наивысший приоритет, поддержку которого он обеспечивает. В других сообщениях управления соединением протокола GSMР поле 654 "приоритет" устанавливается на ноль. Передающийся узлом и ингибируется приемным узлом.

Сообщение "добавить переход" является сообщением управления соединением протокола GSMР, используемым для установления соединения виртуального канала или для добавления дополнительного ветви к существующему соединению виртуального канала. Соединение определяется полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI", в выходная ветвь определяется полем 642 "выходной порт", полем 646 "выходной VPI" и полем 648 "выходной VCI", причем приоритет соединения определяется полем 654 "приоритет". Кроме того, сообщение "добавить переход" может использоваться для проверки состояния соединения, запущенного в ATM коммутаторе. На фиг.13а показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "добавить переход" протокола GSMР от контроллера коммутации. На этапе 600 контроллер коммутации передает сообщение запроса "добавить переход" протокола GSMР, которое принимается ATM коммутатором. ATM коммутатор на этапе 602 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI" принятого сообщения запроса "добавить переход". Если такое виртуальное соединение не существует, то на этапе 604 ATM коммутатор устанавливает соединение, как определено в сообщении запроса "добавить переход". Если коммутатор на этапе 622 определяет, что такое виртуальное соединение существует, то на этапе 668 ATM коммутатор определяет, существует ли в коммутаторе выходная ветвь,

полем 646 "выходной VPI" и полем 648 "выходной VCI" принятого сообщения запроса "добавить переход". Если определено, что такой выходной ветви нет, то ATM коммутатор на этапе 668 добавляет новую выходную ветвь, как определено в сообщении запроса "добавить переход". После этапа 664 или 668 коммутатор на этапе 670 определяет, осуществлена ли указанная операция успешно. Если операция была неуспешной, то ATM коммутатор на этапе 672 посылает контроллеру коммутации сообщение ответа "добавить переход", которое является копией сообщения запроса "добавить переход" с сообщением ответа "добавить переход" с полем 626 "результат", указывающим собой, что операция не была успешной. Если операция была успешной, то ATM коммутатор на этапе 674 проверяет поле 626 "результат" сообщения запроса "добавить переход", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AskAll, то ATM коммутатор на этапе 676 посылает ответ с индикацией успешной операции контроллеру коммутации. Сообщение ответа "добавить переход" является копией сообщения запроса "добавить переход" с полем 626 "результат", указывающим успешный результат. Если коммутатор на этапе 666 определяет, что выходная ветвь, определенная в сообщении запроса "добавить переход", уже существует, то коммутатор на этапе 680 проверяет, является ли приоритет, определенный в поле "приоритет" 654 сообщения запроса, отличающимся от текущего приоритета данной выходной ветви. Если коммутатор определил, что запрашиваемый приоритет отличается от текущего приоритета, то коммутатор на этапе 682 изменяет приоритет выходной ветви на тот, который определен в сообщении запроса "добавить переход". Если приоритеты одинаковы, то коммутатор не изменяет приоритет (показано этапом 684).

Сообщение "удалить переход" является сообщением управления соединением протокола GSMР, используемым для удаления соединения виртуального канала, или в случае последней ветви для удаления соединения виртуального канала. Соединение определяется полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI", в выходная ветвь определяется полем 642 "выходной порт", полем 646 "выходной VPI" и полем 648 "выходной VCI". На фиг.13b показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "удалить переход" протокола GSMР от контроллера коммутации. На этапе 690 контроллер коммутации передает сообщение запроса "удалить переход" протокола GSMР, которое принимается ATM коммутатором. ATM коммутатор на этапе 692 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "входной порт", полем 638 "входной VPI" и полем 640 "входной VCI" принятого сообщения запроса "удалить переход". Если такое виртуальное соединение существует, то на этапе 622 определяет, что такое виртуальное соединение существует, то на этапе 668 ATM коммутатор определяет, существует ли в коммутаторе выходная ветвь,

на этапе 694 ATM коммутатор определяет, существует ли в коммутаторе выходной ветвь, как определено в сообщении запроса "Удалить дерево". После этапа 714 коммутатор на этапе 716 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то ATM коммутатор на этапе 718 проверяет поле 626 "результат" сообщения запроса "Удалить дерево", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AskAI, то ATM коммутатор на этапе 720 посылает ответ с индикацией успешной операции удаления дерева. Сообщение ответа "Удалить дерево" с индикацией успеха является копией принятого сообщения (этап 732 определяет, что соединение, определенное в сообщении запроса "Удалить дерево", не существует, или коммутатор на этапе 724 определяет, что операция удаления передает сообщение ответа "Удалить дерево" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "Удалить дерево" с индикацией сбоя является копией принятого сообщения (этап 736 определяет, что операция проверки не удалась, то коммутатор на этапе 746 устанавливает действительный номер ветвей в поле 650 "число ветвей" сообщения ответа "проверить дерево" с индикацией сбоя и передает его к контроллеру коммутации с полем 628 "результат", указывающим сбоя, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код".

Сообщение "проверить дерево" используется для проверки числа ветвей соединения виртуального канала. Соединение определяется полем 634 "выходной порт", полем 638 "выходной VPI" и полем 640 "выходной VCI". После 642 "выходной порт", поле 646 "выходной VPI" и поле 648 "выходной VCI" в сообщении "проверить дерево" не используются, устанавливаются в ноль контроллером коммутации и информируются коммутатором. На фиг.13с показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "проверить дерево" протокола GSNP от контроллера коммутации. На этапе 730 контроллер коммутации передает сообщение запроса "проверить дерево" протокола GSNP, которое принимается ATM коммутатором. ATM коммутатор на этапе 732 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "выходной порт", полем 638 "выходной VPI" и полем 640 "выходной VCI". Если коммутатор на этапе 732 определяет, что такое соединение виртуального канала существует, то на этапе 734 ATM коммутатор проверяет действительное число ветвей для конкретного соединения и сравнивает действительное число с тем, которое указано в поле 650 "число ветвей" принятого сообщения запроса "проверить дерево". Если указанные числа совпадают, то указанная операция осуществляется успешно. Если операция была успешной, то ATM коммутатор на этапе 738 проверяет поле 626 "результат"

дерево), как определено в сообщении запроса "Удалить дерево". После этапа 714 коммутатор на этапе 716 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то ATM коммутатор на этапе 718 проверяет поле 626 "результат" сообщения запроса "Удалить дерево", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AskAI, то ATM коммутатор на этапе 720 посылает ответ с индикацией успешной операции удаления дерева. Сообщение ответа "Удалить дерево" с индикацией успеха является копией принятого сообщения (этап 732 определяет, что соединение, определенное в сообщении запроса "Удалить дерево", не существует, или коммутатор на этапе 724 определяет, что операция удаления передает сообщение ответа "Удалить дерево" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "Удалить дерево" с индикацией сбоя является копией принятого сообщения (этап 736 определяет, что операция проверки не удалась, то коммутатор на этапе 746 устанавливает действительный номер ветвей в поле 650 "число ветвей" сообщения ответа "проверить дерево" с индикацией сбоя и передает его к контроллеру коммутации с полем 628 "результат", указывающим сбоя, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код".

Сообщение "проверить дерево" используется для проверки числа ветвей соединения виртуального канала. Соединение определяется полем 634 "выходной порт", полем 638 "выходной VPI" и полем 640 "выходной VCI". После 642 "выходной порт", поле 646 "выходной VPI" и поле 648 "выходной VCI" в сообщении "проверить дерево" не используются, устанавливаются в ноль контроллером коммутации и информируются коммутатором. На фиг.13с показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "проверить дерево" протокола GSNP от контроллера коммутации. На этапе 730 контроллер коммутации передает сообщение запроса "проверить дерево" протокола GSNP, которое принимается ATM коммутатором. ATM коммутатор на этапе 732 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "выходной порт", полем 638 "выходной VPI" и полем 640 "выходной VCI". Если коммутатор на этапе 732 определяет, что такое соединение виртуального канала существует, то на этапе 734 ATM коммутатор проверяет действительное число ветвей для конкретного соединения и сравнивает действительное число с тем, которое указано в поле 650 "число ветвей" принятого сообщения запроса "проверить дерево". Если указанные числа совпадают, то указанная операция осуществляется успешно. Если операция была успешной, то ATM коммутатор на этапе 738 проверяет поле 626 "результат"

чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AskAI, то ATM коммутатор на этапе 740 посылает ответ с индикацией успешной операции проверки дерева контроллеру коммутации. Сообщение ответа "проверить дерево" с индикацией успеха является копией принятого сообщения запроса "проверить дерево" с полем 626 "результат", указывающим успешный результат. Если на этапе 738 определено, что ответ в случае успеха не требуется, то коммутатор не выдает ответного сообщения (этап 742). Если коммутатор на этапе 732 определяет, что соединение, определенное в сообщении запроса "проверить дерево", не существует, то коммутатор на этапе 744 передает сообщение ответа "проверить дерево" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "проверить дерево" с индикацией сбоя является копией принятого сообщения запроса "проверить дерево" с полем 626 "результат", указывающим сбоя, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код". Если коммутатор на этапе 736 определяет, что операция проверки не удалась, то коммутатор на этапе 746 устанавливает действительный номер ветвей в поле 650 "число ветвей" сообщения ответа "проверить дерево" с индикацией сбоя и передает его к контроллеру коммутации с полем 628 "код", установленным в ноль.

Сообщение "удалить все" используется для удаления всех соединений на входном порте коммутатора. Входной порт определяется полем 634 "выходной порт". В сообщении "удалить все" поле 638 "выходной VPI", поле 640 "выходной VCI", поле 642 "выходной порт", поле 646 "выходной VPI" и поле 648 "выходной VCI" не используются, устанавливаются в ноль контроллером коммутации и информируются коммутатором. На фиг.13с показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "удалить все" протокола GSNP от контроллера коммутации. На этапе 750 контроллер коммутации передает сообщение запроса "удалить все" протокола GSNP, которое принимается ATM коммутатором. ATM коммутатор на этапе 752 определяет, существует ли какое-либо соединение на входном порте коммутатора, как определено полем 634 "выходной порт". Если коммутатор на этапе 752 определяет, что такое соединение существует, то на этапе 754 ATM коммутатор удаляет все соединения для входного порта коммутатора, как определено в принятом сообщении запроса "удалить все". Затем коммутатор на этапе 756 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то ATM коммутатор на этапе 758 проверяет поле 626 "результат" сообщения запроса "удалить все", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает AskAI, то ATM коммутатор на этапе 760 посылает ответ с индикацией успешной операции удаления

"удалить все" с индикацией успеха является копией принятого сообщения запроса "удалить все" с полем 626 "результат", указывающим успешный результат. Если на этапе 758 определено, что ответ в случае успеха не требуется, то коммутатор не выдает ответного сообщения (этап 762). Если коммутатор на этапе 752 определяет, что нет соединений для удаления, то коммутатор на этапе 764 "удалить все" к коммутатору на этапе 764 передает сообщение ответа "удалить все" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "удалить все" с индикацией сбоя является копией принятого сообщения запроса "удалить все" с полем 626 "результат", указывающим сбоя, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код".

Сообщение "переместить корень" представляет собой сообщение управления соединением протокола GSNP, используемое для перемещения всего дерева виртуальных соединений с текущего входного порта, входного VPI и входного VCI к новому входному порту, входному VPI и входному VCI. На фиг.13с показана структура сообщения 770 "переместить корень", иллюстрирующая соединение протокола GSNP, используемого в качестве запроса и ответа. Сообщение 770 "переместить корень" управления соединением протокола GSNP содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле 622 "версия", 8-битовое поле 624 "тип сообщения", 8-битовое поле 626 "результат", 32-битовое поле 628 "код", 32-битовое поле 630 "идентификатор операции", 32-битовое поле 632 "номер сеанса порта", 32-битовое поле 772 "старый входной порт", 4-битовое поле 774 "установленное на ноль", 12-битовое поле 776 "старый входной VPI", 16-битовое поле 778 "старый входной VCI", 8-битовое поле 780 "новый входной порт", 8-битовое поле 782 "установленное на ноль", 12-битовое поле 784 "новый входной VPI", 16-битовое поле 786 "новый входной VCI", 32-битовое поле 624 "тип сообщения", поле 626 "результат", поле 628 "код", поле 630 "идентификатор операции", поле 632 "номер сеанса порта", используются принципиально тем же путем, как и для других сообщений управления соединением протокола GSNP, как рассмотрено выше. Разрешено поле 788 не устанавливаться, устанавливается на ноль передатчиком и игнорируется приемником. В сообщении "переместить корень" текущее соединение виртуального канала определяется полем 772 "старый входной порт", полем 778 "старый входной VCI", а новое соединение виртуального канала определяется полем 780 "новый входной порт", полем 784 "новый входной VPI" и полем 786 "новый входной VCI". На фиг.13с показана обобщенная диаграмма, иллюстрирующая работу ATM коммутатора, который принимает сообщение запроса "переместить корень" протокола GSNP от контроллера коммутации. На этапе 790 контроллер коммутации передает сообщение запроса "переместить корень" протокола

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

коммутатором. АТМ коммутатор на этапе 792 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 772 "старый входной порт". Поле 776 "старый входной VCI" и поле 778 "старый выходной VPI" и поле 780 "новый входной VCI" принятого сообщения запроса "переместить ветвь". Если коммутатор на этапе 792 определяет, что такое соединение виртуального канала существует, то на этапе 794 АТМ коммутатор определяет ли соединение виртуального канала, являясь ли соединением виртуального канала, определенное полем 780 "новый входной порт", полем 784 "новый входной VPI" и полем 786 "новый входной VCI", не назначенным. Если на этапе 794 определено, что такое соединение виртуального канала является не назначенным, то коммутатор на этапе 796 перемещает каждую выходную ветвь существующего соединения виртуального канала для установления нового соединения виртуального канала, как определено в сообщении запроса "переместить ветвь". После этапа 796 коммутатор на этапе 798 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то АТМ коммутатор на этапе 800 проверяет поле 626 "результат" сообщения запроса "переместить ветвь", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает ACKALL, то АТМ коммутатор на этапе 802 посылает ответ с индикацией успешной операции перемещения корня контроллеру коммутации. Сообщение ответа "переместить ветвь" с индикацией успеха является копией принятого сообщения запроса "переместить ветвь" с полем 626 "результат", указывающим успешный результат. Если на этапе 800 определено, что ответ в случае успеха не требуется, то коммутатор не вырабатывает ответного сообщения (этап 804). Если коммутатор на этапе 792 определяет, что строю соединение, определенное в сообщении запроса "переместить ветвь", не существует, то коммутатор на этапе 806 передает сообщение ответа "переместить ветвь" с индикацией сбоя контроллеру коммутации с указанием соответствующего кода сбоя. Сообщение ответа "переместить ветвь" с индикацией сбоя является копией принятого сообщения запроса "переместить ветвь" с полем 626 "результат", указывающим сбой и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код". Если коммутатор на этапе 794 определяет, что новое соединение виртуального канала, определенное в сообщении запроса "переместить ветвь", является назначенным, то коммутатор не производит никаких изменений в существующих соединениях и устанавливает поле 628 "код" в ноль в сообщении ответа "переместить ветвь" с индикацией сбоя (этап 808) перед передачей его контроллеру коммутации на этапе 808. Еще одно сообщение управления соединением протокола GSNP сообщение "переместить ветвь" используется для перемещения одной выходной ветви соединения виртуального канала с ве текущего выходного порта, выходного VPI и выходного VCI к новому выходному порту, новому VPI и новому VCI в том же самом

показана структура сообщения 820 "переместить ветвь" управления соединением протокола GSNP, используемого в качестве запроса и ответа. Сообщение 820 "переместить ветвь" управления соединением протокола GSNP содержит (в порядке от старшего бита к младшему биту) следующие поля: 8-битовое поле 622 "версия"; 8-битовое поле 624 "тип сообщения"; 8-битовое поле 626 "результат"; 8-битовое поле 628 "код"; 32-битовое поле 630 "идентификатор операции"; 32-битовое поле 632 "номер сеанса порта"; 32-битовое поле 634 "входной порт"; 4-битовое поле 636, установленное на ноль; 12-битовое поле 640 "входной VCI"; 32-битовое поле 822 "старый выходной порт"; 8-битовое поле 824, установленное на ноль; 12-битовое поле 826 "старый выходной VPI"; 16-битовое поле 828 "старый выходной VCI"; 32-битовое поле 830 "новый выходной порт"; 32-битовое поле 832, установленное на ноль; 12-битовое поле 834 "новый выходной VCI"; 24-битовое резервное поле 838, поле 940 "приоритет". Поле 622 "версия", поле 624 "тип сообщения", поле 626 "результат", поле 628 "код", поле 630 "идентификатор операции", поле 632 "номер сеанса порта", используются принципиально тем же путем, как и для других сообщений управления соединением протокола GSNP, как рассмотрено выше. Резервное поле 838 не используется, устанавливается на ноль передатчиком и инпринируется приемником. Поле 940 "приоритет" используется аналогичным образом, как описано выше для поля 654 "приоритет" сообщения управления соединением протокола GSNP. В сообщении "переместить ветвь" соединение виртуального канала определяется полем 634 "входной порт", полем 636 "входной VPI", полем 640 "входной VCI". Старая ветвь соединения виртуального канала определяется полем 822 "старый выходной VPI", полем 826 "старый выходной VPI", полем 828 "старый выходной VCI". Новая ветвь соединения виртуального канала определяется полем 830 "новый выходной порт", полем 834 "новый выходной VPI" и полем 836 "новый выходной VCI". На фиг. 13 показана обобщенная диаграмма, иллюстрирующая работу АТМ коммутатора, который принимает сообщение запроса "переместить ветвь" протокола GSNP от контроллера коммутации. На этапе 842 контроллер коммутации передает сообщение запроса "переместить ветвь" протокола GSNP, которое принимается АТМ коммутатором. АТМ коммутатор на этапе 844 определяет, существует ли в коммутаторе соединение виртуального канала, как определено полем 634 "входной порт", полем 636 "входной VPI", полем 640 "входной VCI". Если коммутатор на этапе 844 определяет, что такое соединение виртуального канала существует, то коммутатор на этапе 846 определяет, существует ли в этом соединении виртуального канала старая выходная ветвь, как определено полем 822 "старый выходной порт", полем 826 "старый выходной VCI", полем 828 "старый выходной VCI".

на этапе 846 определено, что такая старая выходная ветвь существует, то коммутатор на этапе 848 добавляет новую выходную ветвь, как определено полем 830 "новый выходной порт", полем 834 "новый выходной VPI" и полем 836 "новый выходной VCI" в сообщении запроса "переместить ветвь" и удаляет старую выходную ветвь, как определено в сообщении запроса "переместить ветвь". После этапа 848 коммутатор на этапе 850 определяет, осуществлена ли указанная операция успешно. Если операция была успешной, то АТМ коммутатор на этапе 852 проверяет поле 626 "результат" сообщения запроса "переместить ветвь", чтобы определить, следует ли передавать ответ в случае успешного запроса. Если поле "результат" сообщения запроса указывает ACKALL, то АТМ коммутатор на этапе 854 посылает ответ с индикацией успешной операции перемещения ветви контроллеру коммутации. Сообщение ответа "переместить ветвь" с индикацией успеха является копией принятого сообщения запроса "переместить ветвь" с полем 626 "результат", указывающим успешный результат. Если на этапе 852 определено, что ответ в случае успеха не требуется, то коммутатор не вырабатывает ответного сообщения (этап 856). Если коммутатор на этапе 844 определяет, что соединение виртуального канала, определенное в сообщении запроса "переместить ветвь", не существует, или если коммутатор на этапе 846 определяет, что старая ветвь, определенная в сообщении запроса "переместить ветвь", не существует, или если коммутатор на этапе 850 определен, что операция перемещения ветви неуспешна, то коммутатор на этапе 858 не модифицирует никаких состояний соединений и передает на этапе 860 сообщение сбоя контроллеру коммутации с индикацией сбоя. Контроллер коммутации с указанием соответствующего кода сбоя. Сообщение ответа "переместить ветвь" с индикацией сбоя является копией принятого сообщения запроса "переместить ветвь" с полем 626 "результат", указывающим сбой, и с типом сбоя, указываемым соответствующим кодом сбоя в поле 628 "код". Обеспечивая управление коммутационными портами, сообщение управления портами протокола GSNP обеспечивает возможность активизации порта для осуществления обслуживания, исключения порта из процедуры обслуживания, закончывания или установки в исходное состояние. На фиг.14 представлена структура сообщения 870 управления портом протокола GSNP, использующего в качестве сообщения запроса и ответа. Сообщение 870 управления портом протокола GSNP может содержаться в поле 546 сообщения протокола GSNP инкапсулированного пакета 540 протокола GSNP, как показано на фиг.11а. Согласно фиг.14, сообщение 870 управления портом протокола GSNP содержит (в порядке от старшего бита к младшему) следующие поля: 8-битовое поле 622 "версия"; 8-битовое поле 624 "тип сообщения"; 8-битовое поле 626 "результат"; 8-битовое поле 628 "код"; 32-битовое поле 630 "идентификатор операции"; 32-битовое поле 872 "порт",

32-битовое поле 876 "номер последовательности события", 8-битовое поле 878 "флаг события", 8-битовое поле 880 "длительность", 16-битовое поле "функция". Поле 622 "версия", поле 624 "тип сообщения", поле 626 "результат", поле 628 "код", поле 630 "идентификатор операции", поле 874 "номер сеанса порта" используются принципиально тем же путем, как и для других сообщений управления соединением протокола GSNP, как рассмотрено выше. Поле 872 "порт" указывает номер порта, для которого применяется сообщение управления портом протокола GSNP. Сообщение управления портом протокола GSNP имеет конкретное поле "тип сообщения" и различные возможные функции, которые могут быть определены в поле 882 "функция". Некоторые из функций GSNP управления портом протокола GSNP включают следующее: функция включения, функция отключения, функция внутреннего запыльцовывания, функция внешнего запыльцовывания, функция двупортного порта, функция сброса слогов событий, каждая коммутационный порт поддерживает номер последовательности события и набор флагов событий (по одному флагу события на каждый тип сообщения "событие". Номер последовательности события устанавливается в ноль, когда порт инициализируется, и получает приращение каждый раз, когда на этом порте обнаруживается асинхронное событие, оповещающее сообщением "событие", независимо от того, передано ли сообщение "событие" или нет. Если коммутационный порт передает сообщение "событие", он устанавливает соответствующий флаг события на данном порте. Тем самым порту "событие" того же самого типа, пока соответствующий флаг события не будет сброшен функцией сброса флагов событий сообщения управления портом протокола GSNP. Использование флагов событий GSNP обеспечивает простое управление потоком, предотвращающее перегрузку контроллера коммутации сообщениями "событие" со стороны коммутатора. В сообщении запроса управления портом протокола GSNP поле 876 "номер последовательности события" не используется, устанавливается на ноль контроллером коммутации и инпринируется коммутатором. В сообщении ответа управления портом протокола GSNP с индикацией успеха поле 876 "номер последовательности события" указывает текущее значение номера последовательности события принятого сообщения запроса управления портом протокола GSNP. В сообщении запроса управления портом протокола GSNP с полем 882 "функция", определяющим сброс флагов событий, конкретные биты в поле 878 сброса соответствующих флагов событий в порте, определенном полем 872 "порт". В сообщении ответа управления портом протокола GSNP с индикацией успеха, в котором поле 882 "функция" определяет сброс флагов событий, биты поля 878 флагов

значения, соответствующие флагам событий для определенного порта, после того как флаги событий, определенные в сообщении запроса, будут сброшены. Путем установки сообщения управления портом протокола GSMR с помощью функции сброса флагов событий, контроллер коммутации способен получить текущее состояние флагов событий и текущий номер последовательности изменения состояния флагов порта без необходимости сброса флагов событий. В других сообщениях управления портом протокола GSMR с другим определенным полем "функция" поля 878 "флаги событий" не используется, устанавливается в ноль контроллером коммутации и индицируется коммутатором. Поле 880 "длительность" используется только в сообщениях управления портом протокола GSMR с полем 882 "функция", определенным как функция внешнего записывания или функция двунаправленного записывания. Поле 880 "длительность" обеспечивает длительность (в секундах), в течение которой сохраняется состояние записывания. По источнику установленного интервала порт, который находился в состоянии записывания, автоматически возвращается в рабочее состояние. В сообщении управления портом с различным определенным полем 882 "функция" поле 880 "длительность" не используется. Оно устанавливается на ноль контроллером коммутации и индицируется коммутатором. В сообщении управления портом поле 882 "функция" определяет действие, которое должно быть осуществлено (определенное действие осуществляется независимо от текущего статуса порта). Функция включения вводит порт в рабочее состояние, а функция отключения выводит порт из рабочего состояния. Функция внутреннего записывания выполняет внутреннее записывание (ATM элементы данных, присоединя на выходной порт с коммутатора, закодированные через входной порт обратно на коммутатор). Функция внешнего записывания выполняет внешнее записывание (ATM элементы данных, присоединя на входной порт из внешней линии связи, закодированные назад в линию связи на физическом уровне, без ввода во входной порт). Функция двунаправленного записывания выполняет как внешнее, так и внутреннее записывание. Функция сброса входного порта устанавливает в исходное состояние входной порт (все соединения, присоединя на контроллер входной порт, удаляются, в операционный средства индицируются, так что все портно индицируются, так что все значения UR/VC/CI для конкретного входного порта в таблице соединений являются незаполненными). Функция сброса флагов событий сбрасывает флаги событий, как описано выше.

Сообщения статистики протокола GSMR обеспечивают возможность контроллеру коммутации запрашивать значения различных счетчиков аппаратных средств, связанных с входными и выходными портами коммутатора и с виртуальными каналами. Предусмотрены

активности виртуального канала и сообщения статистики порта и виртуального канала. Сообщение активности виртуального канала используется для определения того, осуществлялась ли передача трафика по виртуальному каналу. Сообщение активности виртуального канала содержит одну или несколько записей активности виртуального канала. Каждая запись активности виртуального канала используется для запроса и возврата в ответ информации активности, относящейся к одному определенному виртуальному соединению. Если коммутатор поддерживает трафик, учитываемый на каждом виртуальном соединении, то текущее значение счетчика трафика для каждого конкретного виртуального соединения возвращается в запись "счет трафика виртуального канала". Текущее значение счетчика трафика сравнивается с предыдущими значениями для каждого определенного виртуального соединения для определения того, было ли виртуальное соединение активно в соответствующий период. Если коммутатор поддерживает определение трафика, присоединяется на виртуальное соединение некоторым иным образом, отличным от подсчета трафика, то результат может указываться для виртуального соединения с использованием поля "флаг". Сообщение активности виртуального канала. Сообщение статистики порта и виртуального канала используются для запроса различных счетчиков, определенных для трафика и ошибок для порта и виртуального канала. Сообщение статистики порта используется для получения статистических данных порта коммутатора, определенного в поле "порт" сообщения, в сообщении статистики виртуального канала используется для получения статистических данных для виртуального канала (определенного в полях VP/VC/CI сообщения) на входном порте коммутатора, определенного в поле "порт" сообщения.

Сообщения конфигурации протокола GSMR позволяют контроллеру коммутации определять функциональные возможности ATM коммутатора в базовом коммутационном блоке. Определены три типа сообщений конфигурации протокола GSMR: конфигурация коммутатора, конфигурация порта, конфигурация всех портов. Сообщения конфигурации протокола GSMR используют различные форматы для сообщения запроса и сообщения ответа поскольку они содержат различную информацию в своих полях. Передаваемые контроллером коммутации к ATM коммутатору, сообщение запроса конфигурации коммутатора, указанных конкретным полем "тип сообщения", запрашивает у ATM коммутатора данные о его глобальной конфигурации. Затем коммутатор возвращает конфигурацию коммутатора, которое включает в себя поля для типа коммутатора и имени коммутатора для ATM коммутатора, а также версию инсталлированного аппаратно-программного обеспечения коммутатора. Тип коммутатора представляется изготовителем коммутатора

изделия, а имя коммутатора может представлять собой 48-битовый адрес IEEE 802 MAC или иную величину, которая является уникальной в операционном контексте коммутатора. Сообщение запроса конфигурации порта имеет свое собственное поле "тип сообщения" и передается контроллером коммутации к ATM коммутатору. Сообщение запроса конфигурации порта запрашивает у коммутатора информацию конфигурации одного порта, который определен в поле "порт" сообщения запроса конфигурации порта. Коммутатор передает конфигурацию коммутации сообщение ответа конфигурации порта с индикацией успеха, которое включает в себя информацию конфигурации для входной и выходной сторон определенного порта. Информация конфигурации в сообщении ответа конфигурации порта с индикацией успеха включает следующие: текущий номер сеанса порта, минимальное значение VP/CI в таблице соединений на входном порте, которое может поддерживаться протоколом GSMR, минимальное значение VCI в таблице соединений на входном порте, которое может поддерживаться протоколом GSMR, максимальное значение VP/CI в таблице соединений на входном порте, которое может поддерживаться протоколом GSMR. Информация конфигурации также включает следующие: скорость передачи элементов данных (скорость ATM-элементов данных в секунду) для порта, текущее состояние (т.е. работоспособен, неработоспособен, недоступен, внутреннее записывание, внешнее записывание, двунаправленное записывание) для порта; тип порта (тип физического интерфейса, передающий для порта, например наземный, SONET STS-3c, при кодировании при 100 Мбс, 4835B, 155.52 Мбс, DS3 при 44.736 Мбс, 4835B кодирование при 155.52 Мбс, 8B10B кодирование при 155.52 Мбс, физический уровень ATM Forum при 25 Мбс или физический уровень ATM Forum при 61 Мбс), и число приоритетов, которое выходной порт может присваивать соединениям виртуального канала. Информация конфигурации обеспечивается со ссылками на запись порта для конкретного порта. Контроллер коммутации передает сообщение запроса конфигурации порта, которое имеет собственное поле "тип сообщения", на ATM коммутатор для запроса информации конфигурации для всех коммутационных портов. Таким образом, сообщение запроса конфигурации всех портов специально не определяет конкретный порт. Коммутатор передает сообщение ответа конфигурации всех портов с индикацией успеха, которое обеспечивает следующие: число записей портов, содержащихся в ответном сообщении, длину в байтах каждой записи порта и записи порта для каждого порта. Запись порта для каждого порта представляет собой ту же самую информацию конфигурации, что и описана выше для сообщения ответа конфигурации порта с индикацией успеха. Разумеется, если число записей портов превышает определенный максимум,

конфигурации всех портов с индикацией успеха, то записи портов могут быть переданы в нескольких ответных сообщениях с индикацией успеха, которые не превышают установленное максимальное значение.

Сообщения событий протокола GSMR обеспечивают возможность для ATM коммутатора информировать контроллер коммутации об определенных асинхронных событиях. Как упоминалось выше, сообщения событий не подтверждаются. Сообщения событий могут иметь разные типы событий в зависимости от конкретного асинхронного события. Различные сообщения событий включают в себя сообщение "событие работоспособности порта", сообщение "событие неработоспособности порта", сообщение "событие недействительных VP/VC/CI", сообщение "событие нового порта", сообщение "событие заблокированного порта". Каждый порт коммутатора поддерживает номер последовательности событий и набор флагов событий (по одному флагу события на каждый тип сообщения). Если порт коммутатора передает сообщение события, он устанавливает соответствующий флаг события на данный порт. Тем самым порт не разрешается передавать сообщение следующего флага события на этот порт, пока соответствующий флаг события не будет сброшен функцией сброса флагов событий GSMR. Использование флагов событий обеспечивает простое управление потоком, препятствуя перегрузке контроллера коммутации сообщениями событий со стороны коммутатора. Номер последовательности события устанавливается на ноль, когда порт инициализируется, и получает присваивание всякий раз, когда асинхронное событие, уведомляемое сообщением события, обнаруживается на данном порте, независимо от того, передано ли сообщение события или нет. Текущий номер последовательности события включен в сообщения события для информирования контроллера коммутации об асинхронных событиях, которые имели место для данного порта, но о которых не было направлено уведомление посредством сообщения события ввиду действия простого механизма управления потоком. Сообщение "событие работоспособности порта" информирует контроллер коммутации, что конкретный порт изменил свое состояние с неработоспособного на работоспособное. Когда порт становится работоспособным, все соединения на его входном порте удаляются (таблица соединений входного порта являются незаполненными), и коммутатор присваивает новый номер сеанса порта. Сообщение "событие неработоспособности порта" информирует контроллер коммутации, что конкретный порт изменил свое состояние с работоспособного на неработоспособное. Если коммутатор способен обнаружить отказ линии связи, то коммутатор передает сообщение "событие неработоспособности порта" для уведомления контроллера коммутации о неспособности линии связи. Если один или более элементов ATM данных покидают на входной порт с меткой VP/VC/CI, которая в данный момент не присвоена соединению виртуального канала, то

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

состояния SYNRCVD 1064 в состоянии A
SYNSENT 1060. Если условие в
удовлетворено, когда узел находится в
состоянии SYNRCVD 1064, то передающий
узел выполняет операцию обновления
равного по полюсному, передает сообщение
ACK протокола флоситы IFMP-C (показано как
1076) и переходит из состояния SYNRCVD
1064 в состояние ESTAB 1088. Если условие
C. удовлетворено, когда узел находится в
состоянии SYNRCVD 1064, то передающий
узел передает сообщение ACK протокола
флоситы IFMP-C (показано как 1084) и
переходит из состояния SYNRCVD 1064 в
состояние ESTAB 1088.

Если узел находится в состоянии ESTAB 1068, то возможны различные переходы. Как показано на фиг.16б, если условие D или H в узле ESTAB 1068, то узел находится в состоянии ESTAB 1068, передающий узел остается в состоянии ESTAB 1068 и передает сообщение RSTACK протокола близости IFMP-C-С (показано как 1068). Если условие F в узле ESTAB 1068, то узел находится в состоянии ESTAB 1068, но узел остается в состоянии ESTAB 1068, но не передает сообщение (показано как 1068). Если условие E в узле ESTAB 1068, то передающий узел передает сообщение SYN протокола близости IFMP-C (показано как 1068) и переходит из состояния ESTAB 1068 в состояние SYNSENT 1060. Если состояние S или G узла ESTAB 1068, то передающий узел передает сообщение ACK протокола близости IFMP-C (показано как 1060) и остается в состоянии ESTAB 1068. Каждый узел на одной стороне линии связи не должен передавать более одного сообщения ACK протокола близости IFMP-C, генерируемого от одного пакета, на период таймута в соответствии с конкретным вариантом осуществления.

Сообщение интерфейса IFMR-C может содержать в поле "содержание" протокола IFMR-C идентифицированный пакет 10000 пакета IFMR-C, посвященного п. 15а. Как упомянуто выше, после того как информация будет получена, становится сообщением интерфейса IFMR-C использоваться для обнаружения и конфигурирования инфраструктуры. В конкретном варианте осуществления сообщения интерфейса IFMR-C включают сообщения запроса перечня интерфейсов и соответствующего ответа, сообщение ошибок перечня интерфейсов, сообщение обращения с запросом к интерфейсу и соответствующего ответа, сообщение ошибок запроса к интерфейсу, сообщения запроса и ответа конфигурирования интерфейса, сообщение ошибок конфигурирования интерфейса. Сообщения перечня интерфейсов и сообщения запроса к интерфейсу протокола IFMR-C имеют специфические значения в поле 1016 "тип сообщения". Разумеется, в других вариантах осуществления возможны дополнительные сообщения или сообщения, выполняющие сходные функции.

Сообщения запроса перечня интерфейсов и соответствующего ответа и сообщения ошибок перечня интерфейсов используются

доступны на ИРМ-С исполнительным устройству. На фиг.17а и 17б представлены структуры сообщений запроса и ответа структуры интерфейсов соответственно. Как показано на фиг.17а, соответствие запроса перечня интерфейсов 1100 имеет обобщенный формат, который был рассмотрен выше со ссылками на фиг.15б, причем поле 1030 "тело сообщения" содержит поле 1112 "следующие "печеные" для (идентификатор, используемый для координации взаимодействия)", представляющей собой 32-битовое значение, возвращаемое из предыдущего ответного сообщения перечня интерфейсов. Поле 1112 "следующий идентификатор" используется для поддержки перечня идентификаторов, которые занимают более одного сообщения. Значение поля 1112 "следующий идентификатор" представляет собой значение, возвращаемое ИРМ-С исполнительным устройством, для обеспечения возможности следующего сообщения запроса осуществлять продолжение перечня интерфейсов из предыдущего завершенного сообщения. Предварительно определенное значение, например, в конкретном варианте осуществления, используется для указания того, что перечень интерфейсов должен запускаться с начала. Если сообщение запроса перечня интерфейсов имеет флаг 1020 "flag", установленный, то ИРМ-С 1000 "flag" и операция уступа, то ИРМ-С возвращает сообщение ответа 1114 перечня интерфейсов с его полем 1020 "flag", (фиг.17б) с его полем 1020 "flag", установленным на флаг ACK, и с полем 1018 "код", установленным на предварительно установленное значение (например, 0 в конкретном варианте осуществления), являющееся отсутствием ошибок. Как показано на фиг.17б, поле 1112 "следующий идентификатор" в сообщении ответа 1114 перечня интерфейсов представляет собой 32-битовое значение, возвращаемое как часть сообщения. Если значение поля 1112 "следующий идентификатор" равно 0, то все интерфейсы перечислены. Если значение не равно 0, то это значение используется в качестве аргумента для следующего сообщения перечня интерфейсов для получения, имеющихся остальных интерфейсов. ИРМ-С исполнительного устройства присылает каждому интерфейсу уникальные 32-битовый идентификатор, который ИРМ-С для ссылки на конкретный протокол ИРМ-С в других из сообщений интерфейсов. Ответное сообщение 1114 перечня интерфейсов перечисляет в поле 1111120 "идентификатор 1 интерфейса", в поле 1111118 "идентификатор 2 интерфейса", и т.д. все идентификаторы для каждого интерфейса на ИРМ-С исполнительном устройстве, которые могут быть перечислены в ответном сообщении.

Если запрос перечня сообщений был неуспешным, то IFMP-C исполнителю необходимо возвращать сообщение об ошибке перечня интерфейсов, которое содержит заголовок протокола IFMP-C. В сообщении об ошибке перечня интерфейсов флаг NACK должен быть установлен в поле 1020 "флаг", поле 1018 "код" должно указывать причину.

должно быть тем же самым, что и в сообщении запроса перечня интерфейсов. В числе причин сбоев, каждая из которых имеет конкретное значение для использования в поле 1018 "note", может быть то, что в поле 1112 был указан недействительный идентификатор. IGRP-с исполнительное устройство не смогло назначить буфер сообщения для завершения ответа, специфическая ошибка пользователя обусловила невозможность завершения запроса и т.д.

Сообщения обращения с запросом к интерфейсу и соответствующего ответа и сообщения с определенными интересами на интерфейс IPMP-C используют структуру, фиг. 17с. Сообщения обращения с запросом к интерфейсу содержат ошибку запроса к интерфейсу использования для получения атрибутов, связанных с сообщением. Сообщение 116 содержит обращение с запросом к интерфейсу и соответствующего ответа. Как видно из фиг. 17с, сообщение 1130 обращения с запросом к интерфейсу имеет общесный формат, как ранее описано со ссылкой на фиг. 15б для поля 1030 "тело сообщения". Содержимое поля 1132 "идентификатор интерфейса", которое представляет собой 32-битовый идентификатор уникального атрибута которого запрашивается. Как описано выше, идентификаторы интерфейсов присваиваются IPMP-C исполняющим устройством и могут быть получены с помощью сообщения перечня интересов. Если сообщение 1130 обращения с запросом к интерфейсу имеет флаг PLEASE_ACK в его поле 1020 "флаги" то операция осуществлена успешно, то IPMP-C исполнительное устройство возвращает в ответ сообщение 1134 ответа на запрос к интерфейсу показано на фиг.17d). В адресе, в поле 1018 "порт назначения" устанавливается флажок ACK, а в поле 1019 "тип" установлено значение, указывающее определенное назначение, удаляющее отсутствующие ошибки. Как показано на фиг. 17c, поле 1030 "тело сообщения" в сообщении 11134 ответа на запрос к интерфейсу также содержит 48-битовое поле 1135 "моя информация" предназначенное для указания имени запрашиваемого интерфейса (в конкретном варианте осуществления может быть использован MAC адрес, 8-битовое поле 1138 "тип интерфейса" значения могут быть определены для различных типов интерфейсов, таких как ATM, Ethernet, FastEthernet, Gigabit Ethernet, FDDI или Asynchronous Transfer Mode) и 8-битовое поле 1140 "типа передачи" (указывает физическую среду интерфейса), например, многоканальное оптоволоконно, скрученная пара категории 5, оптоволоконное волокно и т.п.). Кроме того, сообщение 1134 ответа на запрос к интерфейсу также содержит 32-битовое поле 1142 "поддерживаемые скорости" и 32-битовое поле 1144 "текущая скорость". Поле 1142 "поддерживаемые скорости" поддерживает различные скорости передачи (например, 10, 25, 100, 155, 622, 1000 Мбит/c (мегабит в секунду) и другие), которые поддерживают запрошенный интерфейс. Для сообщений передач, поддерживающих более одной скорости передачи, различные флаги могут быть установлены в поле 1144 flags

установлен в поле 1142 "поддерживаемые скорости", если интерфейс поддерживает автоматическое согласование устойчивых скоростей. Поле 1144 "текущая скорость" указывает текущую скорость передачи для запрашиваемого интерфейса. Если интерфейс находится в режиме автоматического согласования, то текущая скорость интерфейса указывается в поле 1144 "текущая скорость" и флаг автоматического согласования установлен в поле 1142 "поддерживаемые скорости". Кроме того, сообщение 1134 ответа на запрос к интерфейсу также содержит 32-битовое поле 114148 "поддерживаемый дуплексный режим" (указывающее скорости дуплексного режима, поддерживаемые запрашиваемым интерфейсом, в том числе в полудуплексном режиме; полностью дуплексный режим или режим автоматического согласования при установлении дуплексного режима; более одного флага может быть установлено для интерфейсов, поддерживающих более одной дуплексной установкой), и 32-битовое поле 1148 "текущий дуплексный режим" (указывающее текущую дуплексную установкой дуплексного режима интерфейса; если интерфейс находится в режиме автоматического согласования, поле 1146 будет указывать соответствующее значение). Кроме того, сообщение 1134 ответа на запрос к интерфейсу также содержит 32-битовое поле 1150 "идентификатор slots интерфейса" (указывает физический slot, который занимает интерфейс на IFMR-используемом устройстве), 32-битовое поле 1152 "идентификатор порта интерфейса" (идентифицирует физический порт, который занимает интерфейс на IFMR-используемом устройстве), 16-битовое поле 1154 "флаги интерфейса" и 16-битовое поле 1158 "статус интерфейса". Поле 1154 "флаги интерфейса" определяет текущие установочной конфигурации на запрашиваемом интерфейсе, причем каждый флаг указывает различные состояние (например, интерфейс приведен в работоспособное состояние, интерфейс находится в смешанном режиме, интерфейс выключен групповые пакеты и т.п.). Поле 1158 "статус интерфейса" указывает текущую информацию статуса (например, фактический статус интерфейса). IFMR-реализует управление протокола IFMR-реализует через данный интерфейс или канал на этом интерфейсе, состояние работоспособности и возможности изменения параметров не подвергается изменению посредством IFMR-контроллера). Кроме того, сообщение 1114134 ответа на запрос к интерфейсу также содержит 32-битовое поле 1158 "минимальная метка приема" и 32-битовое поле 1160 "максимальная метка приема" (указывающие минимальную и максимальную метку передачу) и 32-битовое поле 1164 "максимальная метка передачи". Если интерфейс является ATM интерфейсом, то поле 1158 "минимальная метка приема" и поле 1160 "максимальная метка приема" соответствуют указанию минимальной и максимальной идентификатор виртуального канала, по которому интерфейс может осуществлять прием. Если интерфейс является ATM приемом, то поле 1162 "минимальная метка передачи" и поле 1164 "максимальная метка приема" соответствуют указанию минимальной и максимальной идентификатор виртуального канала, по которому интерфейс может осуществлять передачу.

максимальный идентификатор виртуального канала, по которому интерфейс может осуществлять передачу. Если запрашиваемый интерфейс является интерфейсом иного типа, то поля 1158, 1160, 1162 и 1164 устанавливаются в нуль.

Если обращение с запросом интерфейса безуспешно, то IFMR-с исполнителем устройства передается сообщение об ошибке. Устройство передает сообщение об ошибке интерфейсу, которое передает сообщение IFMR-запросов, имеющий флаг NACK, установленный в поле 1020 "длина", соответствующую причину сбоя указанию, содержащему поле 1018 "код", причем поле 1022 "идентификатор операции", то же самое, что и в сообщении 1130 обращения с запросом, в котором могут быть указаны в поле 1018 "код" сообщения об ошибке. Сообщение об ошибке, передаваемое интерфейсу, включает следующий идентификатор интерфейса, неспособный идентифицировать IFMR-исполнительного устройства, назначить ему сообщение для завершения ответа, специфическая полярность ответа, специфическая полярность ответа и т.п.

Сообщения запроса и ответа конфигурации интерфейса и сообщения об ошибках конфигурации используются для обеспечения возможности изменить конфигурацию интерфейса на ИФМ-С в исполнительном устройстве. На фиг.17 показана структура сообщений 1170 запроса конфигурации интерфейса ИФМ-С к исполнительному устройству. Как показано на фиг.17е, сообщение 1170 запроса конфигурации интерфейса имеет общесистемный формат, как описано в разделе 6.1. Сообщение 1170 запрашивает информацию об устройстве, описанное со ссылками на стр.15 б, д, е, г, а именно поле 1030 "тепло сообщения" идентификатора интерфейса 1132 ("идентификатор интерфейсов" в уникальном образцов идентификации устройств), конфигурацию которого можно изменить; 16-битовое поле 1172 "числа значений"; 16-битовое поле 1174 "установка флагов"; 32-битовое поле 1176 "состояние"; 32-битовое поле 1178 "дуплексный режим". Поле 1172 состоит из слов "указывает, какие флаги ИФМ-С исполнительного устройства должны быть очищены на определенном этапе очистки, чтобы изменить операцию очистки флагов интерфейса. Примерами очистки флагов могут служить следующие: перевод интерфейса в неработоспособное состояние, предотвращение приема из смешанного режима, изменение типа передачи данных, изменение скорости передачи и т.д. Поле 1174 "установка флагов" указывает, какие флаги ИФМ-С исполнительного устройства должны быть установлены на определенном этапе, чтобы изменить операцию очистки. Примерами установки флагов могут служить

следующие: перевод интерфейса с рабочего состояния, ввод параметров режима, обеспечение возможности приема интерфейсом данных от группы передатчиков (табл. 1). Поле 1176 "скорость" используется для обеспечения возможности ИРМ-С контролировать изменение скорости (из числа поддерживаемых интерфейсом скоростей: 10, 25, 50, 100, 150, 315, 625, 1000 Мбит/с и другие) передатчика интерфейса. Поле 1178 "дуплексный режим" используется для

контроль изменить установку дуплексного режима (из числа поддерживаемых интерфейсов установок дуплексного режима, в том числе автоматически согласованного дуплексного, полуавтоматического или полностью автоматического дуплексного режима) определяющего значение поля 1178 "Скорость" на поле 1178 "Дуплексный режим" на предвещающей дуплексное значение, например, на ноль в конкретном варианте осуществления, указывает, что установка скорости или дуплексного режима не может быть изменена. Если сообщение 1170 запроса конфигурации интерфейса от IFMP-контроллера имеет флаг `PLEASE_ACK` в его поле 1020 "Флаг", и операция успешно осуществлена, то IFMP-исполнительное устройство возвращает в ответ сообщение ответа конфигурации интерфейса, состоящее из поля IFMP-С заголовка (все поля, показанные на фиг.15b, кроме поля "Тело сообщения"), с флагом `ACK`, установленным в его поле 1020 "Флаг", и полем 1018 "код", установленным на предвещающее значение, указывающее отсутствие ошибки.

Если сообщение запроса к конфигурации интерфейса безуспешно, то IFMR-С выполняет следующие действия: возвращает в исполнительное устройство конфигурирующий ответ сообщение об ошибках конфигурации интерфейса, состоящее из IFMR-С заголовка. В сообщении об ошибках перечня интерфейсов флаг NACK должен быть установлен в поле 1020 "tagid", поле 1018 "cod" должно указывать причину сбоя, а поле 1022 "идентификатор операции" должно быть тем же самым, что и в сообщении запроса перечня интерфейсов. Возможные причины сбоя, которые могут быть указаны в поле 1018 "cod", включают следующие:

недействительный идентификатор
интерфейса, использование
недействительных параметров конфигурации,
специфическая пользовательская ошибка,
препятствующая завершению теста и т.п.

Помимо сообщений протокола близости IFMC-C и интерфейса, другие типы сообщений 1012 протокола IFMC-C включают сообщения запроса перехода, соответствующее сообщение ответа и сообщение ошибок. Существуют шесть типов сообщений переходов протокола IFMC-C, используемых для создания

модифицирования, исключения ветвей для пересылки. Более конкретно упомянутые сообщения переходов протокола IFMR-С включают следующие: "добавить переход", "удалить переход", "переместить переход", "удалить дерево", "получить статистику дерева". Каждая из ветвей пересылки имеет значение предопределения. Если имеет место множество совпадений, то используется ветвь с самым низким предопределением.

Как упомянуто выше, основная пересылка осуществляется с помощью автайв, причем поступающая автайв состоит из двух компонентов: входные данные и выходные данные. Выходные данные обеспечивают достаточно информации для обеспечения того, чтобы поступающая автайв могла быть согласована с поступающими автайвами, а выходные данные обеспечивают информацию, необходимую для пересылки автайв для каждого установленного соплавления данных. Основные операции,

состояния пересылки, ИМР-С "добавить переход" и "удалить переход". Добавление более одного перехода (ветви) при одних и тех же входных данных приводит к получению дерева, которое может быть использовано для пересылки входящего пакета к множеству адресатов. Например, первое сообщение "добавить переход", имеющее конкретные входные данные и некоторые выходные данные, устанавливает одностороннюю пересылку пакета. Второе сообщение "добавить переход", имеющее те же самые входные данные и другие выходные данные, преобразует одностороннюю пересылку в многостороннюю пересылку путем добавления другой выходной ветви, связанной с теми же самыми входными данными. Другие выходные ветви могут быть добавлены тем же самым способом с использованием последующих сообщений "добавить переход". Сообщение "удалить переход" для многостороннего соединения с двумя ветвями удаляет ветвь, имеющую определенные выходные данные, преобразуя многостороннюю пересылку в одностороннюю пересылку. Еще одно сообщение протокола ИМР-С, сообщение "удалить ветвь", используется для удаления всех ветвей, совместно используемых оми и те же выходные данные. Второе сообщение протокола ИМР-С, сообщение "переместить переход", используется для изменения существующей выходной информации для существующей ветви. Дополнительные сообщения протокола ИМР-С: сообщение "получить статистику дерева", используемое для получения статистики для периодов заголов, входные данные которых определены, и сообщения "считать переход", используемые для целей диагностики и отладки, позволяют ИМР-С контролировать получаемые данные обо всех ветвях пересылки для данного ИМР-С исполнительного устройства.

Содержание "добавки переход" сообщения запроса "удалить протокол" представляют собой сообщения формата 1200 сообщения (но имеют разные типы сообщений), локализованы на фиг. 18а. Как показано на фиг. 18а, формат 1200 сообщения запроса "добавить/удалить переход" протокола IFMP-C имеет общепринятый формат, как описано выше со ссылкой на фиг. 15б с полям 1030 "тело сообщения", поле 15б с полям 1030 "тело сообщения", поле 1201, включающим следующие поля: поле 1201, идентификатор входного интерфейса, 16-битовое поле 1202 "выходное представление", 16-битовое поле 1204 "входные флаги", 32-битовое поле 1206 "идентификатор выходного интерфейса", 24-битовое резервное поле 1208, 8-битовое поле 1210 "длина ключа", 8-битовое поле 1212 "длина выходного заголовка", 8-битовое поле 1214 "длина удаления", 8-битовое поле 1216 "тип преобразования", 32-битовое поле 1222 "обработка частота обслуживания", поле 1224 "данные ключа обслуживания", поле 1225 "маска входного ключа", предвзятительно определенной ключ, поле 1228 "выходные данные заголовка", предвзятительно определенные данные.

преобразования".

Поле, представленные на фиг.18а (иные, чем показанные на фиг.15 поля обобщенного заголовка протокола IFMP-C), описаны ниже. Привисенное IFMP-C исполнителем устройством и полученное с помощью сообщений переноса интерфейсов протокола IFMP-C поле "идентификатор обобщенного интерфейса" 1201 уникальным образом идентифицирует конкретный входной интерфейс, к которому относится входная ветвь. Поле 1202 "входное предшествование", в конкретном варианте осуществления представляющее собой 16-битовое число без знака, обозначает предшествование, присвоенное ветви. При сравнении входящих пакетов с входными ключами ключ с самым низким предшествованием сравнивается первым. Если более одного элемента имеют одинаковое предшествование, то IFMP-C используетное устройство может выбрать любую из сравниваемых ветвей для пересылки пакета. Поле 1204 "входные флаги" используется для входной ветви, и флаги, указанные в этом поле, используются для обозначения конкретного действия, которое должно быть предпринято, если пакет согласован с этим передающим элементом. Примерами такого действия, обозначенных такими флагами, могут служить следующие: "передача управления вниз", поиск на следующем уровне предшествования после передачи пакета, вместо завершения; "отбрасывание" и игнорирование всех пакетов, которые согласованы с этим входным элементом. Поле 1206 "идентификатор выходного интереса" 1206 "идентификатор выходного интереса" уникальным образом идентифицирует интерес, предназначенный для использования для передачи пакета. Разное поле 1208, которое может быть зашифровано для использования в будущем, в конкретном варианте осуществления может быть установлено передающим элементом на нullo, и игнорируется приемным элементом, если это поле не используется. Поле 1210 "длина ключа" представляющее собой в конкретном варианте осуществления 8-битовое число, которое устанавливает длину поля 1224 "данные входного ключа" и поля 1226 "массив выходного ключа" в байтах. Поле 1212 "длина выходного заголовка", представляющее собой в конкретном варианте осуществления 8-битовое число без знака, определяет длину поля 1228 "данные выходного заголовка" в байтах. Поле 1214 "длина заголовка", представляющее собой 8-битовое число без знака, определяет число байтов для удаления от начала пакета, прежде чем применять преобразование, указанное в поле 1230 "данные преобразования".

Поле 1216 "Тип преобразования" определяет тип модификации (например, не модифицировать, преобразовать IP-пакет в IPv6, пакет типа 1, IP-пакет в IPv6, пакет типа 2, GMP-пакет типа 1 в IP-пакет, IPv6-пакет типа 2 в IP-пакет, стандартная IP-IP-пересылка, усечение пакета или иные изменения), которое должно быть осуществлено перед передачей пакета. Некоторые типы преобразования требуют использования данных, которые не являются

фиг. 15b, но без поля "тепло сообщения"), с флагом NACK, установленным в его поле 1020 "флаг", и с полем 1018 "ход", установленным на предвещающее определенное значение, указывающее причину сбоя (в качестве которых могут быть следующие: неадекватный идентификатор интерфейса, длина входного ключа больше, чем максимальная поддерживаемая длина, недостаточные ресурсы в IFMP-C, исполнительное устройство, выходное преобразование не поддерживается или не расположено, не существует ветвь с входным ключом, как определено для исходной ветви в запросе; исходная ветвь не существует, и ни одна ветвь не соплагуется с новой ветвью; другая ветвь существует с тем же входным ключом, как определено, но с другим флагом, параметры качества обслуживания неадекватны или не поддерживаются, специфическая пользовательская ошибка препятствует завершению запроса и т.п.).

Еще одно сообщение протокола IFMP-C, сообщение "получить статистику дерева", используется IFMP-C контроллером для определения того, когда узлы больше не используются, так чтобы эти узлы можно было восстановить. IFMP-C исполнительное устройство поддерживает счетчик, ведущий отсчет постоянно, пока узел используется для пересылки пакета. Такой узел сохраняется для каждого дерева, так как каждая ветвь такого дерева будет использоваться одно и то же число раз. Как видно из фиг. 19a, сообщение "получить статистику дерева" протокола IFMP-C, обобщенный формат, ранее описанный со ссылками на фиг. 15b, причем поле 1030 "тепло сообщения" включает перечень информации, включенной в себя в конкретном варианте осуществления данные дерева: поле 1402 "данные 1 дерева", поле 1402 "данные 2 дерева". Дополнительные поля "данные дерева" могут быть включены в перечень, содержащийся в поле 1030 "тепло сообщения". На фиг. 19b показана структура 1408 поля "данные дерева", в которой используются все поля данных дерева. Каждая структура поля данных дерева включает следующие поля: поле 1201 "идентификатор входного интерфейса", поле 1202 "выходное преобразование", поле 1204 "входные флаги", 40-битовое резервное поле 1408, поле 1210 "длина ключа", 18-битовое поле 1410 "размер записи", 64-битовое поле 1412 "отсчет использования", поле 1224 "входные данные ключа", предвещающее определенной длины и поле 1225 "маска входного ключа" предвещающее определенной длины. Поле 1201 "идентификатор входного интерфейса" уникальным образом определяет входной интерфейс, к которому относится входная ветвь (для которой должна быть получена статистика дерева). Резервное поле 1408 зарезервировано для использования в будущем, может быть установлено в нуль передатчиком и при этом будет игнорироваться приемником. Поле 1410 "размер записи", которое показывает размер конкретной записи дерева (например, в поле 1402 "данные 1 дерева"), используется для отсчисления начала следующей записи дерева (например, в поле 1404 "данные 2 дерева"),

представляет собой 64-битовое целое число без знака, которое указывает прикреплене всякий раз, когда IFMP-C исполнительное устройство использует конкретное дерево для пересылки пакета. В запросном сообщении поле 1412 "отсчет использования" устанавливается в нуль передатчиком и при этом будет игнорироваться приемником. Остальные поля не описаны здесь, поскольку они были описаны выше со ссылками на фиг. 18a.

Если сообщение запроса 1400 "получить статистику дерева" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаг", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "удалить дерево" протокола IFMP-C, которое имеет тот же самый формат, что и сообщение запроса 1400 (как показано на фиг. 19b), с флагом ACK, установленным в его поле 1020 "флаг", и с полем 1018 "ход", установленным на предвещающее определенное значение, указывающее отсутствие ошибки. Ответное сообщение "получить статистику дерева" также возвращает в ответ значения соответствующих счетчиков в полях "отсчет использования" полей "данные дерева".

Если сообщение запроса "получить статистику дерева" безуспешно, то IFMP-C исполнительное устройство возвращает соответствующее сообщение ошибки, которое представляет собой поля запроса протокола IFMP-C (как показано на фиг. 15b, но без поля "тепло сообщения" сообщения запроса, с флагом NACK, установленным в его поле 1020 "флаг", и с полем 1018 "ход", установленным на предвещающее определенное значение, указывающее причину сбоя (в качестве которых могут быть следующие: неадекватный идентификатор интерфейса, одна из определенных выходных ветвей не существует в IFMP-C исполнительном устройстве; другая ветвь существует с тем же входным ключом, как определено, но с другим флагом, и т.п.).

Еще одним сообщением протокола IFMP-C является сообщение "читать переход", которое используется для целей диагностики и отладки, для обеспечения возможности IFMP-C контроллеру получить данные для всех ветвей пересылки в IFMP-C исполнительном устройстве. Для нумерации всех ветвей в IFMP-C исполнительном устройстве сообщение "читать переход" использует операцию "получить следующий элемент". Всякий раз, когда соответствующий элемент готов, IFMP-C исполнительное устройство возвращает информацию ветви, а также соответствующий идентификатор.

Используемый в качестве аргумента для следующей операции считывания. Этот идентификатор является непременным для IFMP-C контроллера и используется IFMP-C исполнительным устройством для напоминания, где оно остановилось при последнем считывании. Предвещающее определенное значение, например 0 в конкретном варианте осуществления, может быть зарезервировано и использовано IFMP-C контроллером для получения исходного элемента ввода в IFMP-C исполнительное устройство. Последующая

будет давать успешный результат до тех пор, пока все таблицы не будут перечислены, после чего IFMP-C исполнительное устройство выдает указание того, что достигнут конечный пункт.

На фиг. 20a и 20b показаны структуры сообщения 1420 запроса считывания ветви и сообщения 1430 ответа считывания ветви протокола IFMP-C соответственно. Как показано на фиг. 20a, сообщение 1420 запроса считывания ветви протокола IFMP-C имеет обобщенный формат, как описано выше со ссылками на фиг. 15b, причем поле 1030 "тепло сообщения" содержит поле 1201 "идентификатор входного интерфейса" и 32-битовое поле 1422 "следующий идентификатор". Поле 1201 "идентификатор входного интерфейса" уникальным образом определяет входной интерфейс, на котором должны считываться ветви. Поле 1422 "следующий идентификатор" является непременным 32-битовым значением, возвращаемым в виде части предыдущего ответного сообщения "читать ветвь". Значение в поле 1422 "следующий идентификатор" представляет состояние поддерживаемое IFMP-C исполнительным устройством, для поддержки отсложения позиции последнего запроса "читать переход". Предвещающее определенное значение, например 0 в конкретном варианте осуществления, используется для указания того, что запрос должен быть осуществлен с начала сдвига.

Если сообщение запроса "читать переход" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаг", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "читать переход" протокола IFMP-C (фиг. 20b) с флагом ACK, установленным в его поле 1020 "флаг", и с полем 1018 "ход", установленным на предвещающее определенное значение, указывающее отсутствие ошибки. Как показано на фиг. 20b, ответное сообщение "читать переход" протокола IFMP-C имеет обобщенный формат, ранее рассмотренный со ссылками на фиг. 15b, причем поле "тепло сообщения" 1030 содержит следующие поля: поле 1201 "идентификатор входного интерфейса", поле 1202 "выходное преобразование", поле 1204 "входные флаги", поле 1206 "идентификатор выходного интерфейса", 24-битовое резервное поле 1432, поле 1210 "длина заголовка", поле 1212 "длина выходного заголовка", поле 1214 "длина давления", поле 1216 "тип преобразования", поле 1218 "длина данных преобразования", поле 1222 "обработка качества обслуживания", поле 1422 "следующий идентификатор", поле 1224 "входные данные ключа" предвещающее определенной длины, поле 1226 "маска входного ключа" предвещающее определенной длины, поле 1228 "данные выходного заголовка" предвещающее определенной длины, поле 1230 "данные преобразования" предвещающее определенной длины. Резервное поле 1432 зарезервировано для использования в будущем, устанавливается в нуль передатчиком и игнорируется приемником. Поле 1422 "следующий идентификатор"

IFMP-C исполнительным устройством, для отслеживания позиции последнего запроса, и используется в качестве ввода в следующее сообщение "читать переход". Остальные поля были описаны выше со ссылками на фиг. 18a. Если запрос "читать период" был неуспешным, то IFMP-C исполнительное устройство возвращает соответствующее сообщение ошибки, которое состоит из заголовка сообщения запроса "читать переход" протокола IFMP-C, за исключением того, что флаг NACK установлен в поле 1020 "флаг", и в поле 1018 "ход" установлено предвещающее определенное значение, указывающее причину сбоя (в качестве которых могут быть следующие: неадекватный идентификатор интерфейса в сообщении, неадекватный следующий идентификатор, IFMP-C исполнительное устройство не обеспечивает выполнения буфера сообщения для завершения ответа, специфическая пользовательская ошибка препятствует завершению запроса и т.п.).

Помимо сообщений протокола IFMP-C, определенных близость, интерфейсы и ветви, сообщения протокола IFMP-C управления сообщения, такие как сообщение сброса. Сообщение сброса протокола IFMP-C используется для повторной инициализации состояния IFMP-C исполнительного устройства, когда IFMP-C контроллер потерял связь с IFMP-C исполнительным устройством или когда IFMP-C контроллер предполагает, что в IFMP-C исполнительном устройстве произошел сбой. Сообщение сброса протокола IFMP-C выдает команду IFMP-C исполнительному устройству сбросить все его состояние с установкой в исходное состояние. После приема сообщения сброса IFMP-C исполнительное устройство удаляет все ветви пересылки и инициализирует интерфейсы без сброса состояния протокола близости IFMP-C. Сообщение сброса состоит из заголовка IFMP-C, показанного на фиг. 15b (без поля "тепло сообщения"), причем поле 1016 "тип сообщения" установлено для обозначения данного сообщения в качестве сообщения сброса. Если сообщение запроса "сброс" протокола IFMP-C имеет флаг PLEASE_ACK, установленный в его поле 1020 "флаг", и операция осуществлена успешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "сброс" протокола IFMP-C, которое состоит из идентичного заголовка IFMP-C, что и в сообщении запроса "сброс", но с флагом ACK, установленным в его поле 1020 "флаг", и с полем 1018 "ход", установленным на предвещающее определенное значение, указывающее отсутствие ошибки. Если сообщение "сброс" неуспешно, то IFMP-C исполнительное устройство возвращает ответное сообщение "сброс" протокола IFMP-C, которое состоит из идентичного заголовка IFMP-C, что и в сообщении запроса "сброс", но с флагом NACK, установленным в его поле 1020 "флаг", и с полем 1018 "ход", установленным на предвещающее определенное значение, указывающее причину сбоя (например, специфическая пользовательская ошибка, препятствующая завершению запроса).

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

"диагностические ошибки", которое указывает место физических ошибок, которые имели место на определенном АТМ интерфейсе. Другой пример поля 1495 "специальная статистика" для интерфейса Ethernet включает: 64-битовое поле 1542 "принятые ошибки ЦИК", которое указывает число пакетов, которые были приняты с ненадлежащей контрольной суммой при проверке с использованием циклического избыточного кода на конкретном интерфейсе; и 64-битовое поле 1544 "конфликты при передаче", которое указывает число конфликтов, имевших место при попытках передать пакеты на конкретном интерфейсе.

Если запрос "статистика интерфейсов" неудачен, то IFMPC исполнителем устройство возвращает сообщение об ошибке "статистика интерфейсов" протокола IFMPC, которое состоит из идентичного заголовка IFMPC. Сообщение об ошибке "статистика интерфейсов" должно быть идентичным заголовку IFMPC в сообщении запроса "информация узла", но с флагом NASC, установленным в его поле 1020 "флаги", и с полем 1018 "код", установленным на предельно определенное значение, указывающее причину сбоя. Причинами сбоя могут быть следующие: недействительный идентификатор интерфейса, соответствующий перечню чисто интерфейсов не существует, числу интерфейсов в сообщении, невозможность выделения IFMPC исполнителем устройством буфера сообщений для завершения отчета, специфическая пользовательская ошибка, препятствующая завершению запроса или иные причины.

4. Заключение

Заявленное изобретение направлено на создание усовершенствованного способа и устройства для передачи пакетов в сети. Следует иметь в виду, что вышеприведенное описание является иллюстративным и не накладывает никаких ограничений. Различные варианты осуществления должны быть очевидными для специалистов в данной области техники, как вытекающие из сведений, приведенных в описании. Для примера изобретение проиллюстрировано в применении к передаче IP-пакетов, обеспечивающих пересылку речевых данных, факсимильных, данных изображений, факсимильных данных и сигналов данных, однако изобретение не ограничивается приведенными видами сигналов. Кроме того, изобретение проиллюстрировано в связи с конкретными компонентами и рабочими скоростями, однако изобретение не ограничивается конкретными указанными значениями параметров. Следует иметь в виду, что различные конкретные примеры типа сообщений, ошибок и т.п. приведены только для конкретных вариантов осуществления, являющихся возможными вариантами, при этом в изобретении могут использоваться и другие дополнительные или отличные признаки или комбинации таких признаков. Объем изобретения должен определяться не на основе информации, приведенной в описании, а только в соответствии с формулой изобретения, совместно с полным объемом эквивалентных средств, на которые распространяются

Формула изобретения:

1. Способ передачи пакетов данных между узлом восходящей линии связи и узлом нисходящей линии связи в сети, причем указанный узел нисходящей линии связи расположен в направлении нисходящей линии связи относительно узла восходящей линии связи, включающий этапы установления по узлу нисходящей линии связи, выполнения приема пакета данных в упомянутом узле нисходящей линии связи, выполнения классификации потока в упомянутом узле нисходящей линии связи по упомянутому пакету данных для определения того, принадлежит ли этот пакет виртуальному каналу, который должен быть передан в узел восходящей линии связи, выбора в упомянутом узле нисходящей линии связи свободной метки, информации упомянутого узла восходящей линии связи о том, что последующие пакеты, принадлежащие к упомянутому определенному потоку данных, должны передаваться с указанной выбранной свободной меткой, при этом этап выполнения классификации потока включает анализ идентификатора потока пакета данных для определения того, принадлежит ли указанный пакет данным к определенному потоку, который должен быть передан в узел восходящей линии связи, причем указанный идентификатор потока содержит определенные поля заголовка из этого пакета данных, и передачи упомянутых последующих пакетов данных, принадлежащих к определенному потоку, прием упомянутых определенных полей заголовка для обеспечения защиты.

7. Способ по п. 6, отличающийся тем, что упомянутые последующие пакеты могут быть восстановлены с использованием идентификатора потока в месте назначения, предназначенного для приема указанного определенного потока.

8. Способ коммуникации потока в первом узле, причем указанный первый узел имеет нисходящую линию связи к второму узлу и включающую линию связи к третьему узлу, классификации потока в первом узле по первому пакету данных для определения того, принадлежит ли этот первый пакет данным к определенному потоку, который должен быть передан в третий узел, выбора первой свободной метки в упомянутом первом узле, информации упомянутого третьего узла о том, что последующие пакеты, принадлежащие к упомянутому определенному потоку, должны передаваться с указанной выбранной свободной меткой, выполнения второго пакета данных для определения того, принадлежит ли этот второй пакет данным к определенному потоку, который должен быть передан в третий узел, выбора второй свободной метки в упомянутом втором узле, информирования упомянутого первого узла о том, что последующие пакеты данных, принадлежащие к упомянутому определенному потоку, должны передаваться с указанной выбранной второй свободной меткой, информирования упомянутого меткой селективной коммуникации упомянутого определенного потока посредством

5

10

15

20

25

30

35

40

45

50

55

60

нисходящей линии связи в сети, причем указанный узел нисходящей линии связи расположен в направлении нисходящего потока относительно узла восходящей линии связи, включающий этапы установления по упомянутому узлу восходящей линии связи, выполнения приема пакета данных в упомянутом узле нисходящей линии связи, выполнения классификации потока в упомянутом узле нисходящей линии связи по упомянутому пакету данных для определения того, принадлежит ли этот пакет виртуальному каналу, который должен быть передан в узел восходящей линии связи, выбора в упомянутом узле нисходящей линии связи свободной метки, информации упомянутого узла восходящей линии связи о том, что последующие пакеты, принадлежащие к упомянутому определенному потоку данных, должны передаваться с указанной выбранной свободной меткой, при этом этап выполнения классификации потока включает анализ идентификатора потока пакета данных для определения того, принадлежит ли указанный пакет данным к определенному потоку, который должен быть передан в узел восходящей линии связи, причем указанный идентификатор потока содержит определенные поля заголовка из этого пакета данных, и передачи упомянутых последующих пакетов данных, принадлежащих к определенному потоку, прием упомянутых определенных полей заголовка для обеспечения защиты.

7. Способ по п. 6, отличающийся тем, что упомянутые последующие пакеты могут быть восстановлены с использованием идентификатора потока в месте назначения, предназначенного для приема указанного определенного потока.

8. Способ коммуникации потока в первом узле, причем указанный первый узел имеет нисходящую линию связи к второму узлу и включающую линию связи к третьему узлу, классификации потока в первом узле по первому пакету данных для определения того, принадлежит ли этот первый пакет данным к определенному потоку, который должен быть передан в третий узел, выбора первой свободной метки в упомянутом первом узле, информации упомянутого третьего узла о том, что последующие пакеты, принадлежащие к упомянутому определенному потоку, должны передаваться с указанной выбранной свободной меткой, выполнения второго пакета данных для определения того, принадлежит ли этот второй пакет данным к определенному потоку, который должен быть передан в третий узел, выбора второй свободной метки в упомянутом втором узле, информирования упомянутого первого узла о том, что последующие пакеты данных, принадлежащие к упомянутому определенному потоку, должны передаваться с указанной выбранной второй свободной меткой, информирования упомянутого меткой селективной коммуникации упомянутого определенного потока посредством

линии связи в нисходящую линию связи, при этом упомянутый определенный поток из восходящей линии связи может передаваться посредством первого узла в нисходящую линию связи.

9. Способ по п. 8, отличающийся тем, что в качестве второго пакета данных используют первый пакет данных.

10. Способ по п. 9, отличающийся тем, что упомянутые первый, второй и третий узлы используют асинхронный режим передачи.

11. Способ по п. 10, отличающийся тем, что упомянутая первая и вторая свободная метка обе содержат идентификатор виртуального маршрута и идентификатор виртуального канала.

12. Способ коммуникации потока в первом узле, причем указанный первый узел имеет нисходящую линию связи к второму узлу и включающую линию связи к третьему узлу, включающий этапы выполнения классификации потока в первом узле по первому пакету данных для определения того, принадлежит ли этот первый пакет данным определенному потоку, который должен быть передан в третий узел, выбора первой свободной метки в упомянутом первом узле, информации упомянутого третьего узла о том, что последующие пакеты данных, принадлежащие упомянутому определенному потоку, должны передаваться с указанной выбранной первой свободной меткой, выполнения второго пакета данных для определения того, принадлежит ли этот второй пакет данным определенному потоку, который должен быть передан в третий узел, выбора второй свободной метки в упомянутом втором узле, информирования упомянутого первого узла о том, что последующие пакеты данных, принадлежащие к упомянутому определенному потоку, должны передаваться с указанной выбранной второй свободной меткой, информирования упомянутого меткой селективной коммуникации упомянутого определенного потока посредством

5

10

15

20

25

30

35

40

45

50

55

60

RU 2 189 072 C 2

RU 2 189 072 C 2

14. Способ по п. 13, отличающийся тем, что упомянутая свободная метка содержит идентификатор виртуального маршрута и идентификатор виртуального канала.

15. Способ по п. 13, отличающийся тем, что упомянутые узлы, восходящей линии связи и нисходящей линии связи используют асинхронный режим передачи.

16. Способ передачи пакетов данных от первого узла ко второму узлу в сети, включающий этап приема первого пакета данных от упомянутого первого узла, выполнения классификации потока по упомянутому первому пакету данных, селективного присвоения свободной метки потоку, связанному с первым пакетом данных, передачи к первому узлу сообщения, указывающего упомянутую свободную метку, и направления последующих пакетов данных из упомянутого потока с использованием канала, указанного упомянутой свободной меткой.

17. Способ по п. 16, отличающийся тем, что дополнительно включает этап установления времени таймера после присвоения упомянутой свободной метки и направления последующих пакетов данных по упомянутому каналу установленного по умолчанию времени таймера.

18. Способ коммутации потока в первом узле, включающий этап приема первого пакета данных упомянутого потока от узла восходящей линии связи, выполнения классификации потока в первом узле по первой плате данных, выбора первой свободной метки для упомянутого узла восходящей линии связи, передачи упомянутому узлу восходящей линии связи сообщения, указывающего упомянутую первую свободную метку, приема от узла нисходящей линии связи сообщения, указывающего вторую свободную метку, и селективной коммутации последующих пакетов упомянутого потока от упомянутого узла восходящей линии связи к упомянутому узлу нисходящей линии связи на основе упомянутых первой и второй свободных меток.

19. Способ по п. 18, отличающийся тем, что упомянутые первый, второй и третий узлы используют асинхронный режим передачи.

20. Базовый коммутационный блок в системе для передачи пакетов данных в сети, содержащий аппаратный контроллер, связанный с упомянутым контроллером, связанный с упомянутым аппаратным средством коммутации, причем контроллер содержит процессор и память и предназначен для управления упомянутым аппаратным средством коммутации, хранящееся на материальном носителе, обеспечивающее возможность базовому коммутационному блоку осуществлять динамический сдвиг между маршрутизацией и коммутацией пакетов для оптимизации пропускной способности линкного графа.

21. Базовый коммутационный блок по п. 20, отличающийся тем, что упомянутое программное обеспечение использует классификацию потока.

22. Базовый коммутационный блок по п. 21, отличающийся тем, что упомянутое

коммутацию на основе асинхронного режима передачи.

23. Базовый коммутационный блок по п. 22, отличающийся тем, что упомянутая классификация потока использует в качестве метки идентификатор виртуального маршрута и идентификатор виртуального канала.

24. Базовый коммутационный блок по п. 22, отличающийся тем, что упомянутое программное обеспечение включает первую подсистему программного обеспечения, предназначенную на контроллере, для осуществления информационного обмена и управления упомянутым аппаратным средством коммутации.

25. Базовый коммутационный блок по п. 24, отличающийся тем, что упомянутое программное обеспечение включает вторую подсистему программного обеспечения, обеспечивающую информационный обмен между двумя упомянутыми базовыми коммутационными блоками и определение формата для сообщений переадресации потока и подтверждений приема.

26. Базовый коммутационный блок по п. 25, отличающийся тем, что базовый коммутационный блок осуществляет локальное принятие решений классификации и решений в ответ на сообщение переадресации.

27. Базовый коммутационный блок по п. 20, отличающийся тем, что упомянутая сеть включает локальную сеть, содержащую компьютеры.

28. Базовый коммутационный блок по п. 24, отличающийся тем, что первая подсистема программного обеспечения включает протокол управления потоком IFMP.

29. Базовый коммутационный блок по п. 28, отличающийся тем, что вторая подсистема программного обеспечения включает протокол, подсистема программного обеспечения включает общий протокол управления коммутацией GSNP.

30. Базовый коммутационный блок по п. 22, отличающийся тем, что упомянутое программное обеспечение включает средство обработки качества обслуживания.

31. Базовый коммутационный блок по п. 21, отличающийся тем, что упомянутое аппаратное средство коммутации использует быстросрабатывающую пакетную технологию.

32. Базовый коммутационный блок по п. 21, отличающийся тем, что упомянутое аппаратное средство коммутации использует технологию ретрансляции кадров.

33. Базовый коммутационный блок по п. 21, отличающийся тем, что упомянутое средство коммутации использует технологию Gigabit Ethernet.

34. Шлюзовый блок коммутации в системе для передачи пакетов в сети, причем упомянутая система включает базовый коммутационный блок, связанный с упомянутым шлюзовым коммутационным узлом посредством линии связи, содержащий контроллер шлюзового блока, содержащий процессор, память и множество плат сетевых интерфейсов, программно обеспечен, хранящееся на материальном носителе, обеспечивающее возможность шлюзовому блоку коммутации переадресовывать поток пакетов данных к упомянутому базовому коммутационному блоку, чтобы осуществлять

коммутацией пакетов для оптимизации пропускной способности пакетного графа.

35. Шлюзовый блок коммутации по п. 34, отличающийся тем, что упомянутое программное обеспечение использует классификацию потока.

36. Шлюзовый блок коммутации по п. 34, отличающийся тем, что упомянутый базовый коммутационный блок использует упомянутую на основе асинхронного режима передачи.

37. Шлюзовый блок коммутации по п. 36, отличающийся тем, что упомянутый шлюзовый блок коммутации и упомянутый базовый коммутационный блок используют в качестве метки идентификатор виртуального маршрута и идентификатор виртуального канала.

38. Шлюзовый блок коммутации по п. 37, отличающийся тем, что упомянутое программное обеспечение включает первую подсистему программного обеспечения, предназначенную на контроллере шлюзового блока, для осуществления информационного обмена между упомянутым шлюзовым блоком коммутации и упомянутым базовым коммутационным блоком в упомянутой системе и определения формата для сообщений переадресации потока и подтверждений приема.

39. Шлюзовый блок коммутации по п. 38, отличающийся тем, что упомянутый шлюзовый блок коммутации осуществляет локальное принятие решений о классификации потока и решений в ответ на сообщение переадресации.

40. Шлюзовый блок коммутации по п. 39, отличающийся тем, что первая подсистема программного обеспечения включает протокол управления потоком IFMP.

41. Шлюзовый блок коммутации по п. 39, отличающийся тем, что упомянутый базовый коммутационный блок использует технологию Gigabit Ethernet.

42. Исполнительное устройство коммутации в системе для передачи пакетов данных в сети, причем упомянутая система включает базовый коммутационный блок, связанный с упомянутым исполнительным устройством коммутации посредством линии связи, в базовый коммутационный блок содержит контроллер и процессор коммутации, содержащий процессор, память и множество плат сетевых интерфейсов, причем конкретная одна из упомянутого множества плат сетевых интерфейсов обеспечивает упомянутую линию связи, а по меньшей мере одна из упомянутого множества плат сетевых интерфейсов имеет возможность связи по меньшей мере с одним узлом сети, и компьютерно-читаемый программный код, хранящийся на материальном носителе, содержащий упомянутую память, носитель, содержащий компьютерно-читаемый программный код, обеспечивающий возможность контроллера базового коммутационного блока осуществлять обмен между упомянутыми исполнительным устройством коммутации и базовым коммутационным блоком в упомянутой системе.

43. Исполнительное устройство коммутации по п. 42, отличающееся тем, что упомянутый компьютерно-читаемый программный код содержит первую подсистему, предназначенную на упомянутом контроллере, и вторую подсистему, предназначенную в памяти исполнительного устройства коммутации, при этом упомянутые первая и вторая подсистемы обеспечивают информационный обмен между упомянутыми исполнительным устройством коммутации и базовым коммутационным блоком в упомянутой системе.

44. Исполнительное устройство коммутации по п. 42, отличающееся тем, что упомянутый компьютерно-читаемый программный код обеспечивает возможность упомянутому контроллеру базового коммутационного блока выдавать команду упомянутому исполнительному устройству коммутации о том, как обрабатывать упомянутые пакеты данных в потоке, принятый от упомянутого процессора коммутации.

45. Исполнительное устройство коммутации по п. 42, отличающееся тем, что упомянутый процессор коммутации использует растровую передачу, ретрансляцию кадров, быстросрабатывающую пакетную коммутацию, Ethernet со скоростью 10 Мбс, Ethernet со скоростью 100 Мбс или Gigabit Ethernet.

46. Исполнительное устройство коммутации по п. 42, отличающееся тем, что по меньшей мере одна из упомянутого множества плат сетевых интерфейсов представляет собой плату сетевого интерфейса сети асинхронного режима передачи.

47. Исполнительное устройство коммутации по п. 42, отличающееся тем, что по меньшей мере одна из упомянутого множества плат сетевых интерфейсов представляет собой плату сетевого интерфейса сети асинхронного режима передачи.

48. Исполнительное устройство коммутации по п. 42, отличающееся тем, что упомянутый компьютерно-читаемый программный код содержит первую подсистему, предназначенную на упомянутом контроллере, и вторую подсистему, предназначенную в памяти исполнительного устройства коммутации, при этом упомянутые первая и вторая подсистемы обеспечивают информационный обмен между упомянутыми исполнительным устройством коммутации и базовым коммутационным блоком в упомянутой системе.

49. Исполнительное устройство коммутации по п. 48, отличающееся тем, что упомянутое исполнительное устройство

узел по отношению к базовому коммутационному блоку, который осуществляет локальное принятие решений о классификации потока и решений в ответ на сообщение передатчика. Устройство по п. 48, отличающееся тем, что

программный код содержит программное обеспечение протокола управления потоком для клиентов IFMP-C.
 Присорбент по пунктам:
 31.01.1996 по пп. 1-23, 27-50;
 22.11.1996 по пп. 24-26.

Таблица 1

Компоненты контроллера	
Микропроцессор	Процессор Intel Pentium, 133 МГц
Системная память	ЗУПВ 16 Мбайт/Кэш 256 К
Материнская плата	Материнская плата Intel Endeavor
Плата АТМ интерфейса	Zeipel PCI ATM NIC (155 Мб/с)
Стационарный или жесткий диск	Диск IDE 500 Мбайт
Накопители	Для стандартных гибких и CD-ROM
Источник питания	Стандартный источник питания
Шасси	Стандартное Шасси

RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

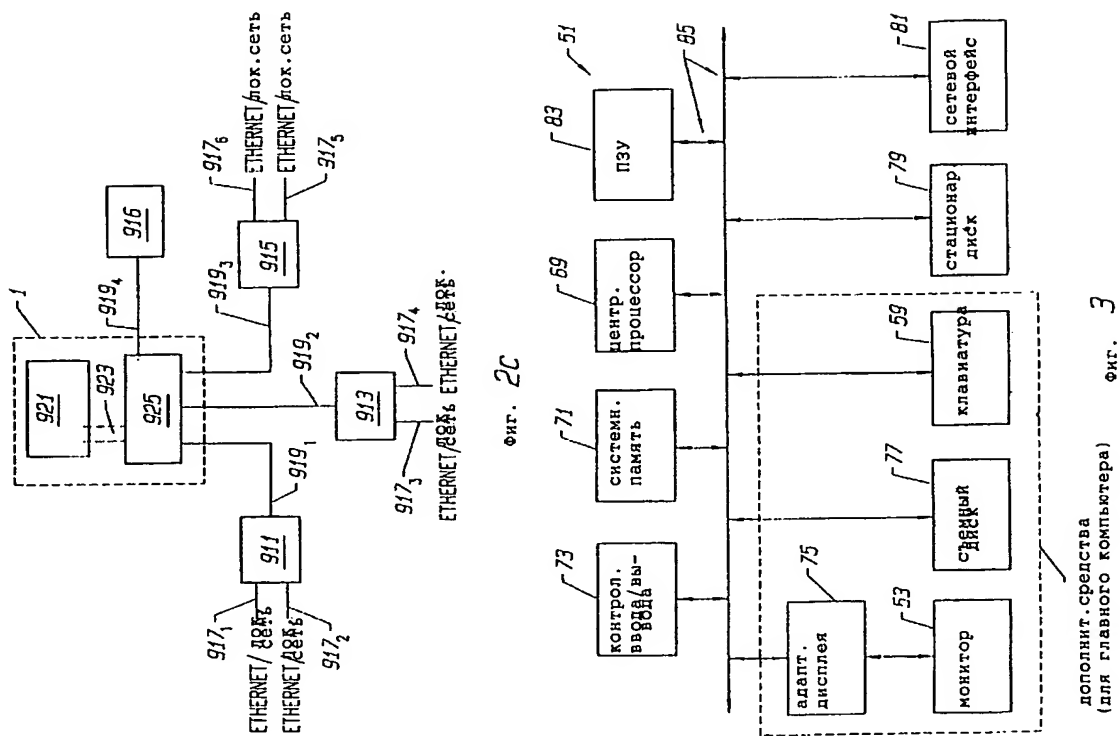
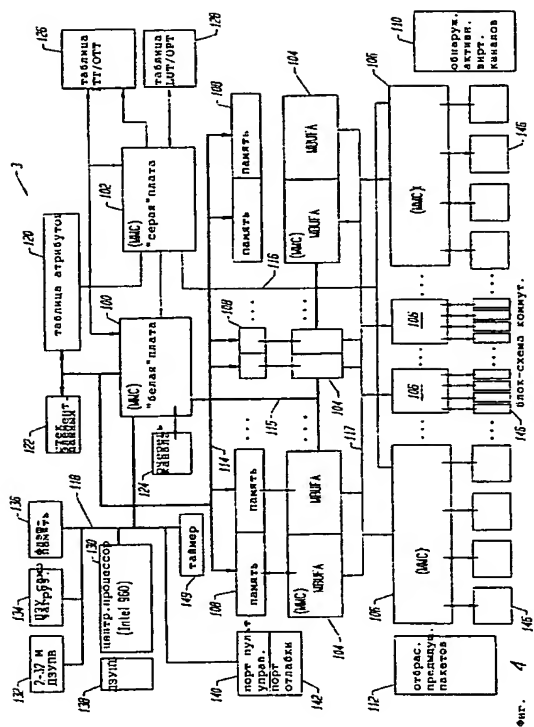
RU 2 1 8 9 0 7 2 C 2

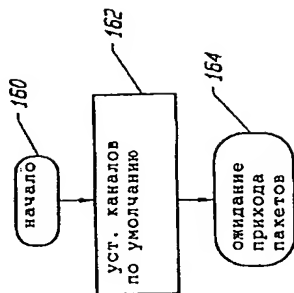
RU 2 1 8 9 0 7 2 C 2

Компоненты коммутатора

Переключающий сердечник	
Комплект микросхем	Комплект микросхем АТМ коммутатора ММС Networks АТMS 2000 (белая микросхема, серая микросхема, микросхемы MBUF, микросхемы PIF)
Память общих данных	Стандартные модули памяти
Счетчики пакетов	Стандартные счетчики
Микроконтроллерный комплекс	
ЦПБ	Intel 960SA/CF/HX
Динамическое ОЗУ	Стандартные модули динамического ОЗУ
ПЗУ	Стандартное ПЗУ
Флэш-память	Стандартная флэш-память
Контроллер динамич.ОЗУ	Стандартные микросхемы FPGА, ASIC и т.д.
Слабонный универсальный асинхронный приемопередающий порт	16552 DUART
Внешний таймер	Стандартный таймер
Подузел приемопередачика	
Физический интерфейс	PMC-Sierra PM5346

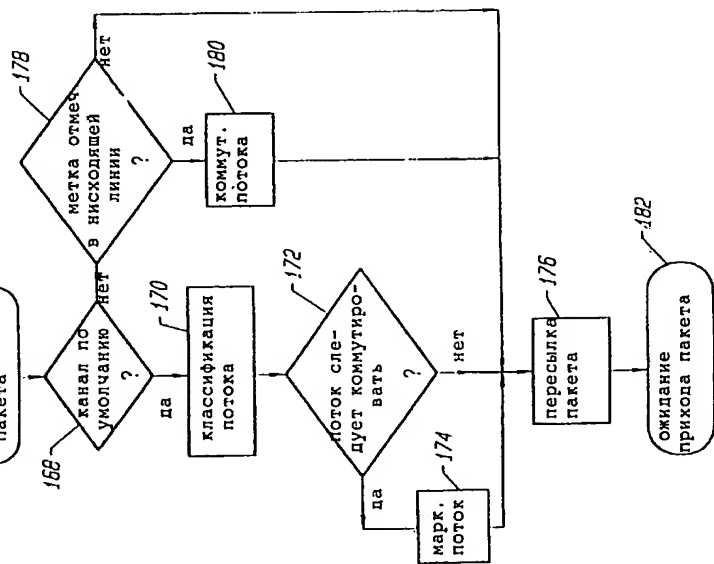






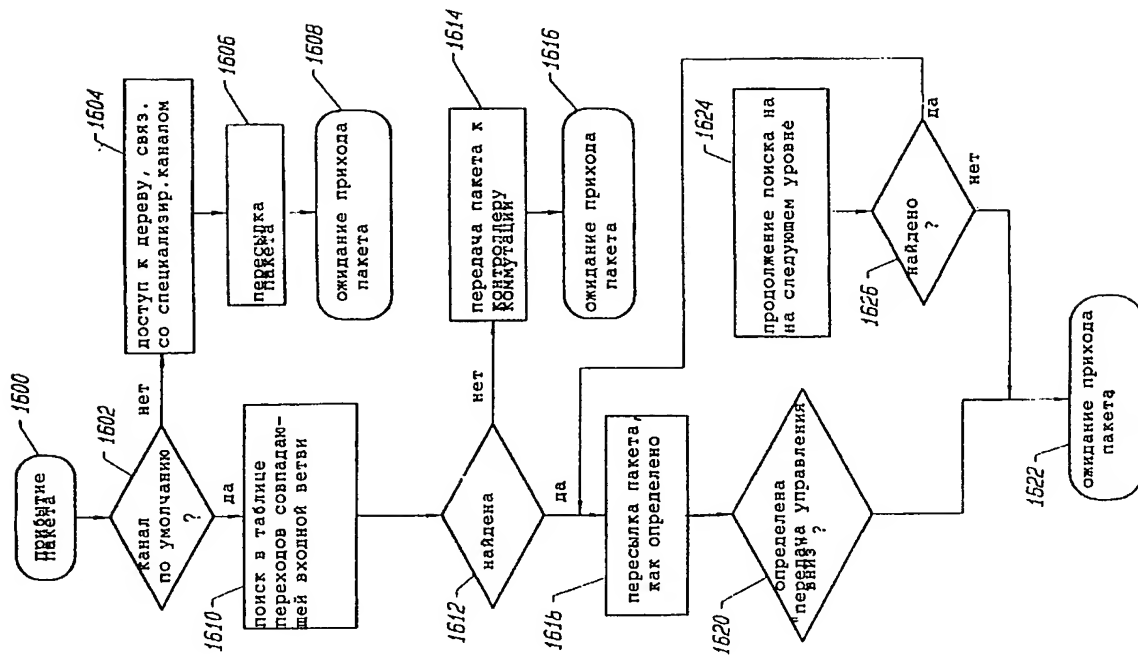
Фиг. 5a

RU 2 189 072 C 2



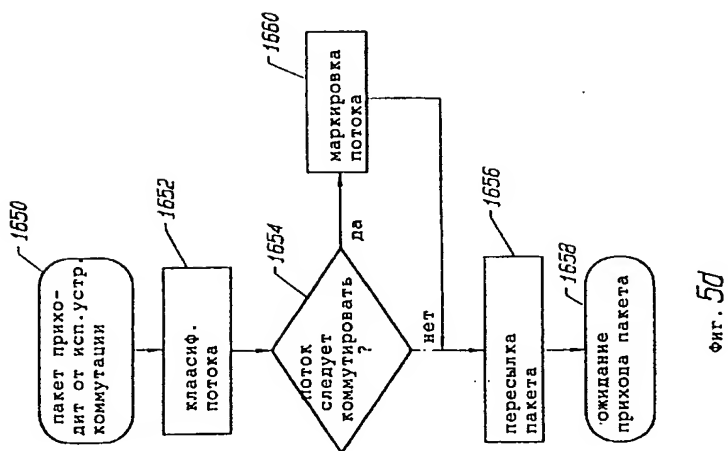
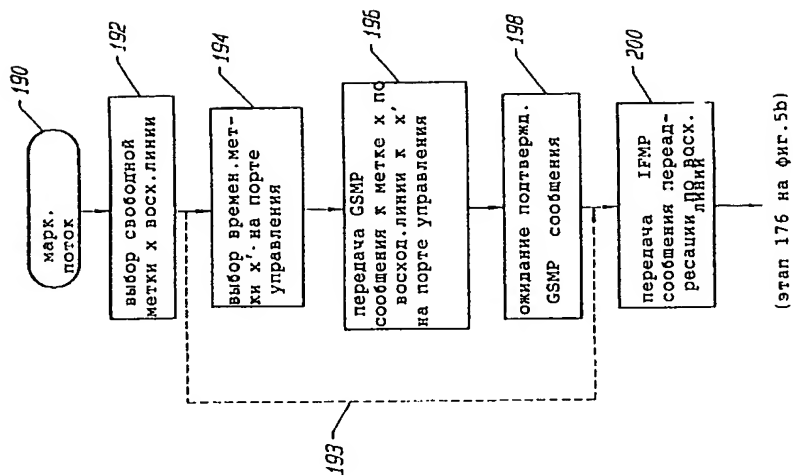
Фиг. 5b

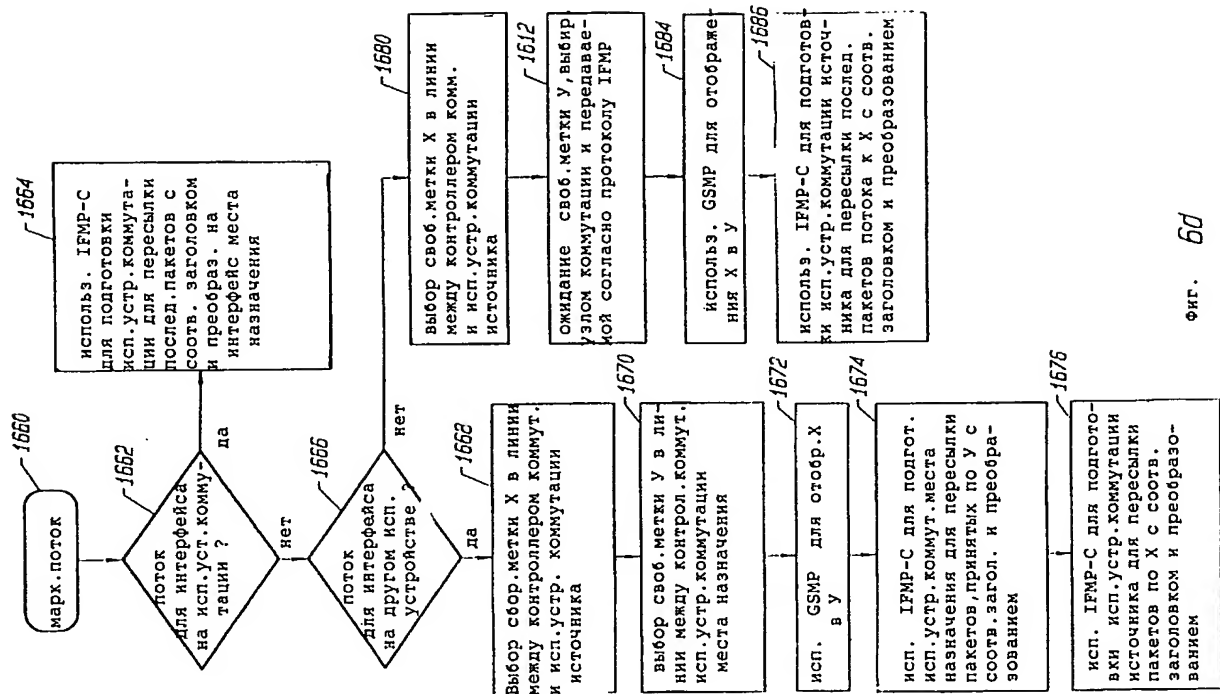
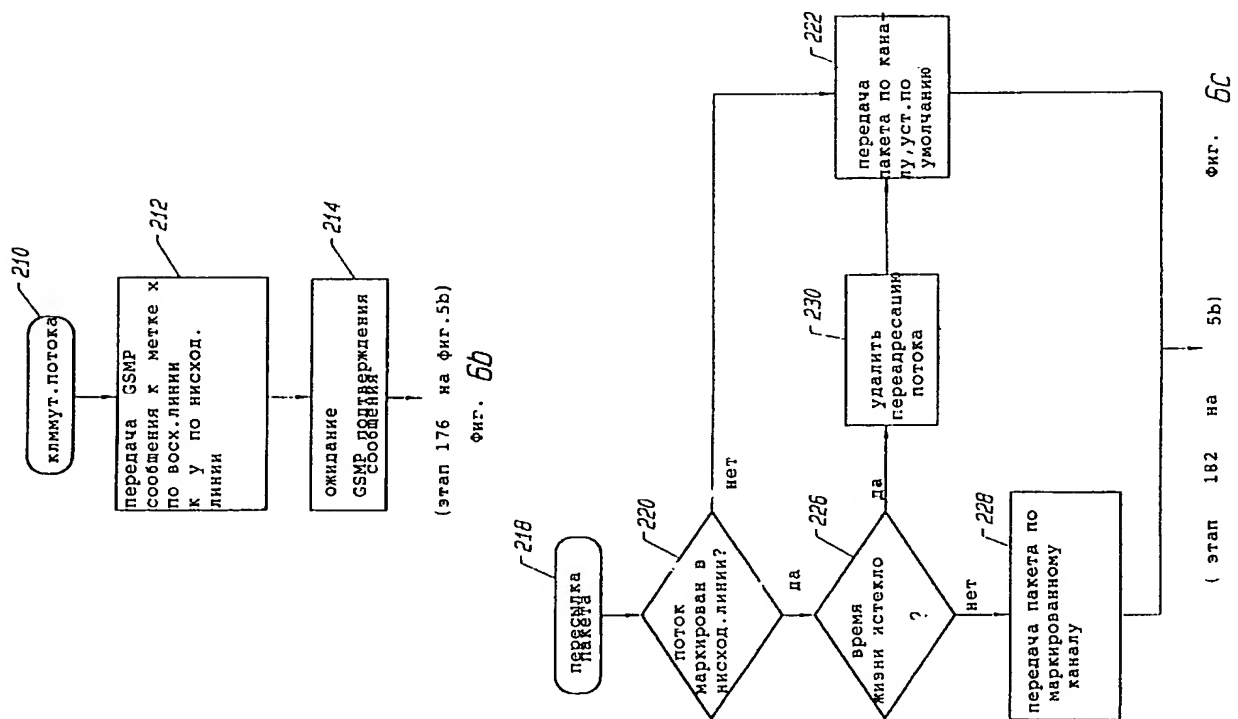
RU 2 189 072 C 2

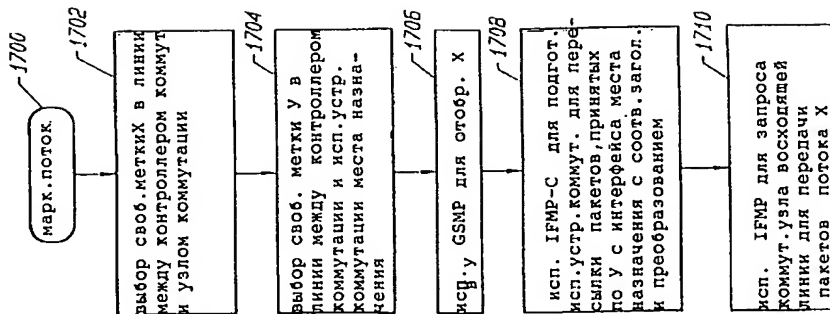


Фиг. 5c

RU 2 189 072 C 2



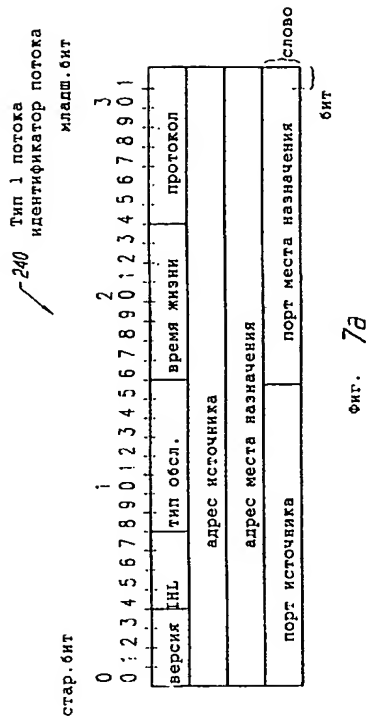




фиг. 6е

RU 2 189 072 C 2

RU 2 189 072 C 2



фиг. 7а

RU 2 189 072 C 2

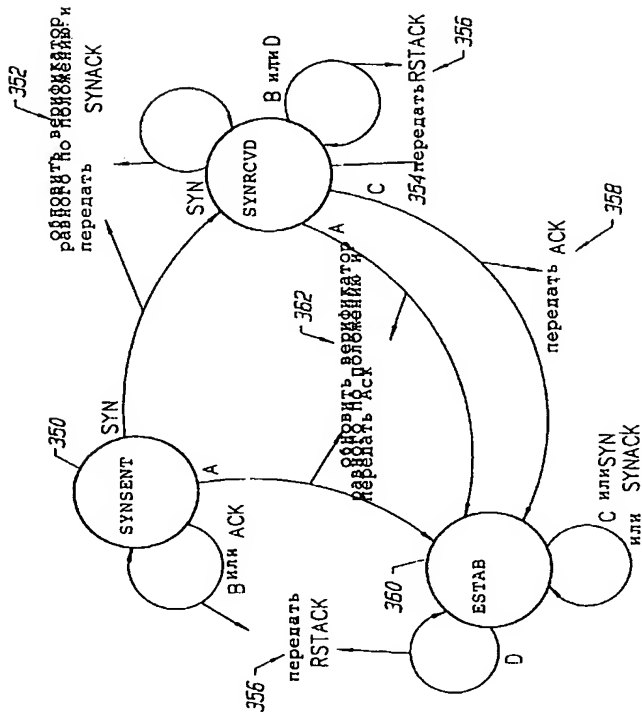
300

Фиг. 88

r (IPv4)

Фиг. 86. 86.

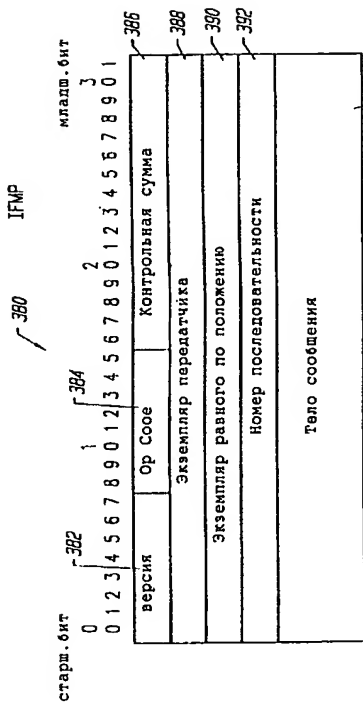




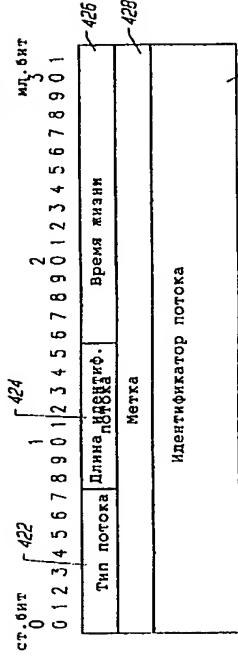
Условия:
 А: SYNACK AND %C
 В: SYNACK AND NOT (%C)
 С: ACK AND %B AND %C
 D: ACK AND NOT (%B AND %C)

Фиг. 8d

RU 2189072 C2

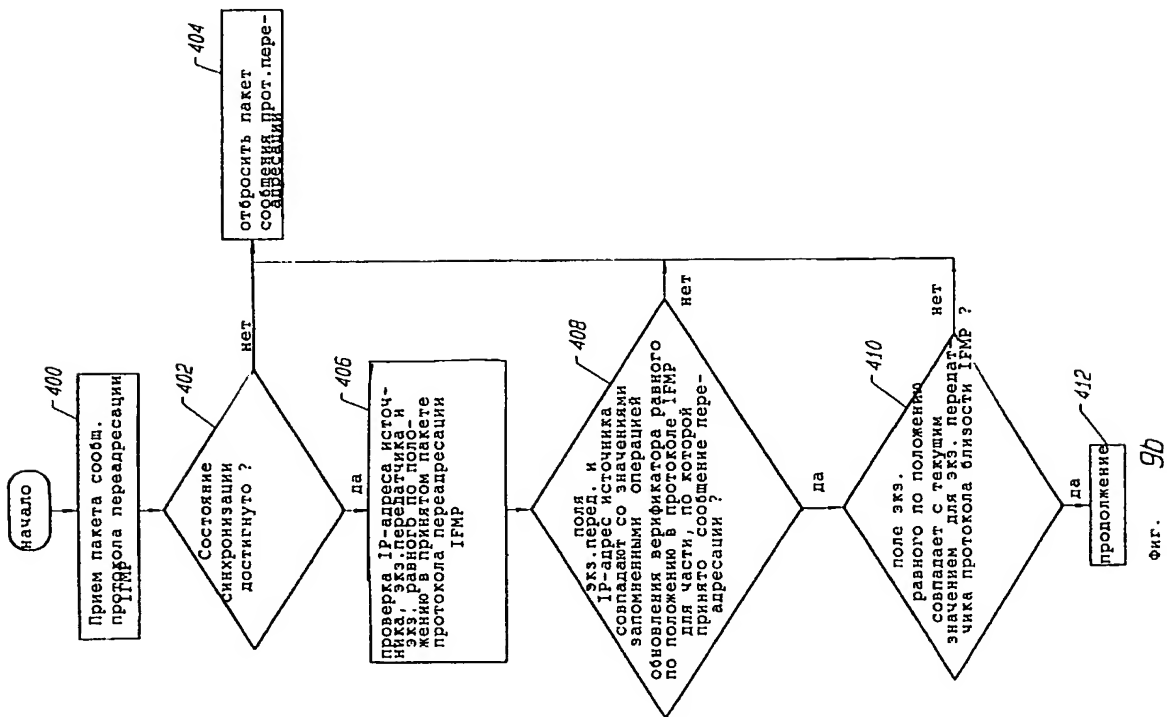


Фиг. 8a



Фиг. 8b

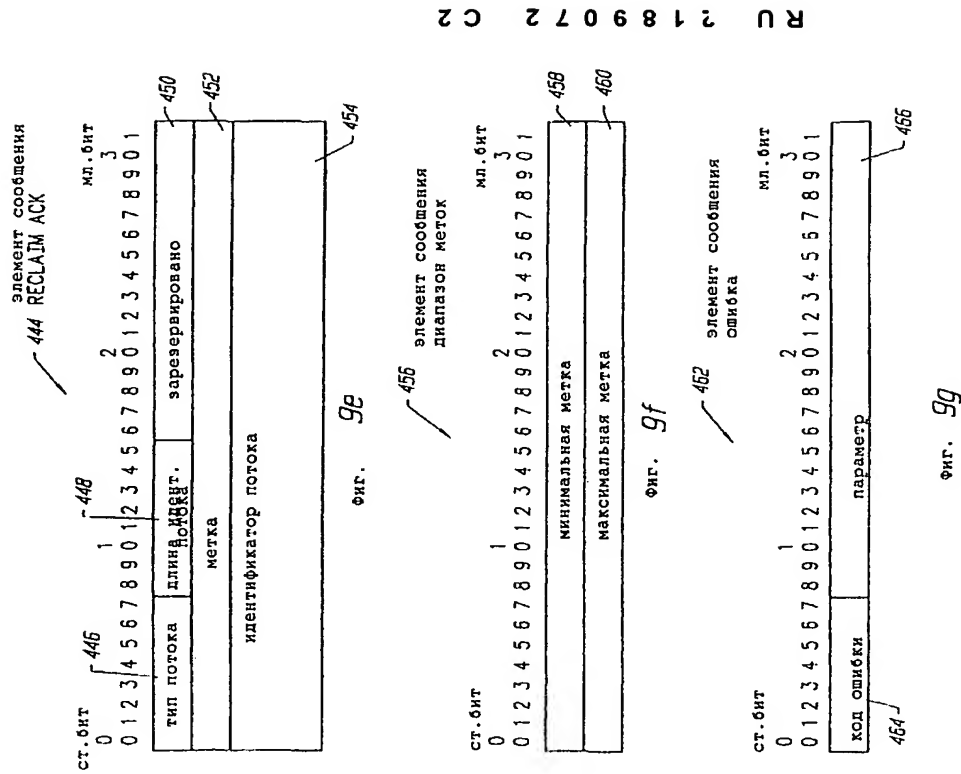
RU 2189072 C2



RU 2 189 072 C 2

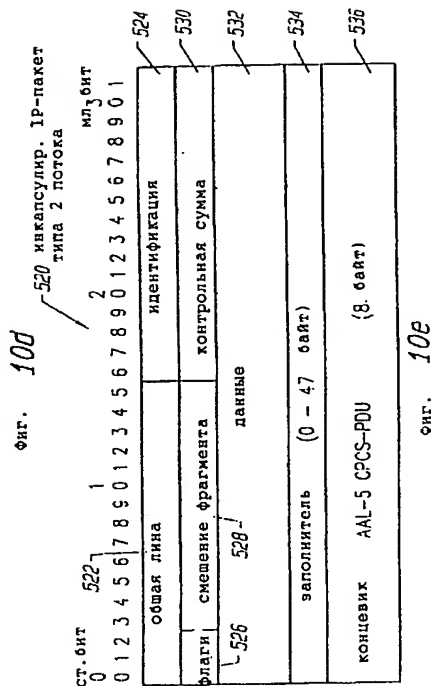
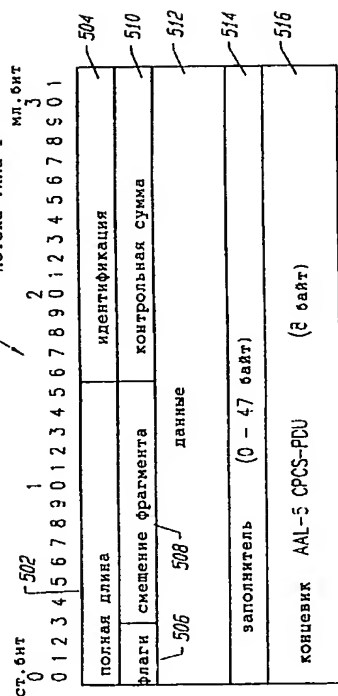
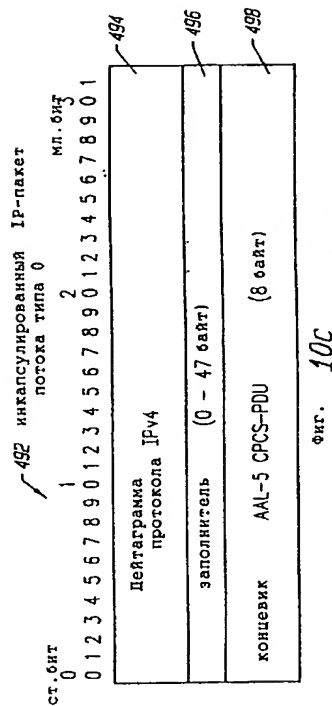
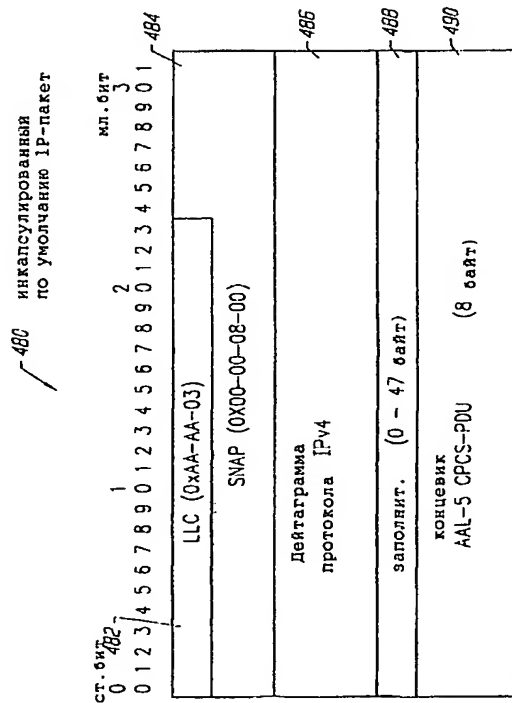
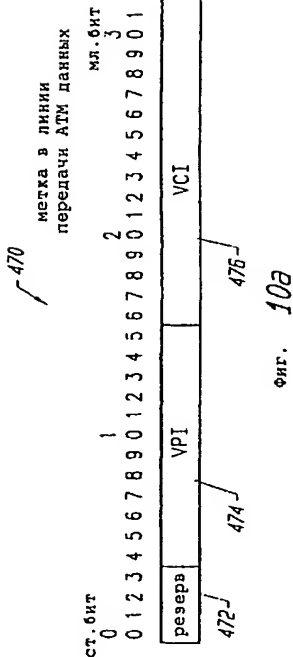
7b

RU 2 189 072 C 2



RU 2 189 072 C 2

7g

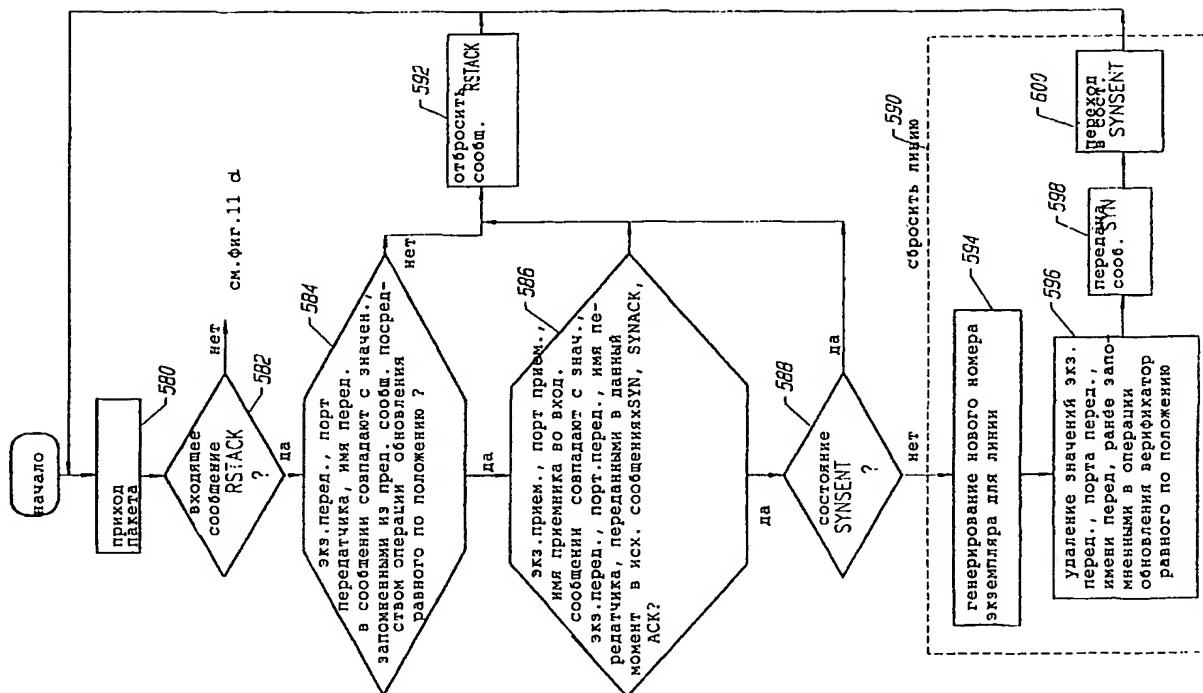
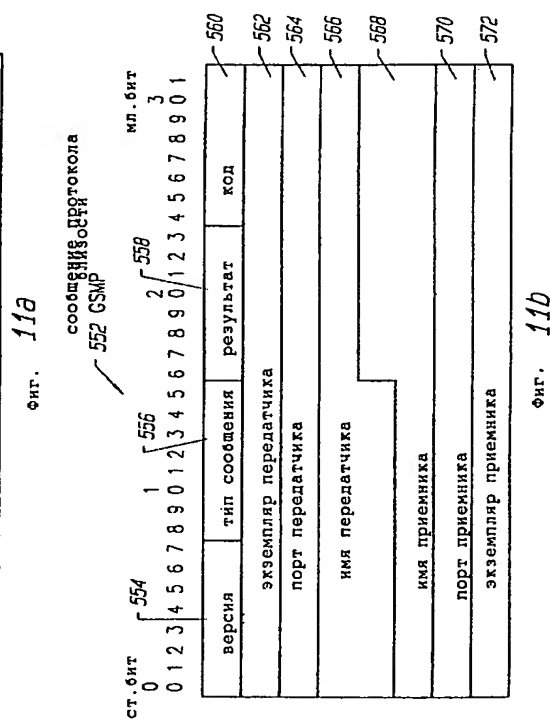
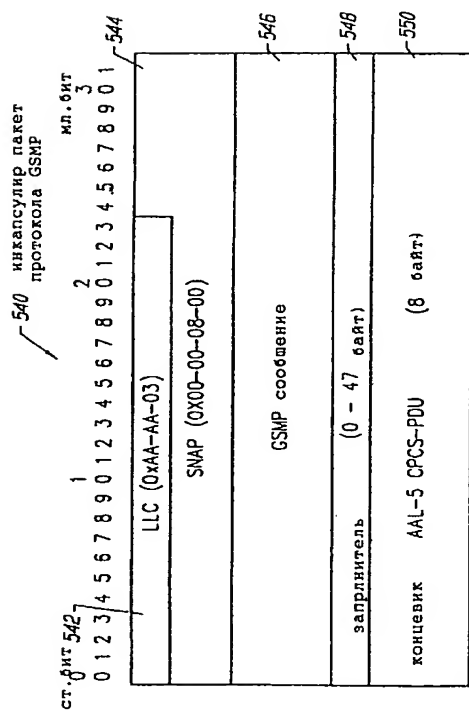


RU 2189072 C2

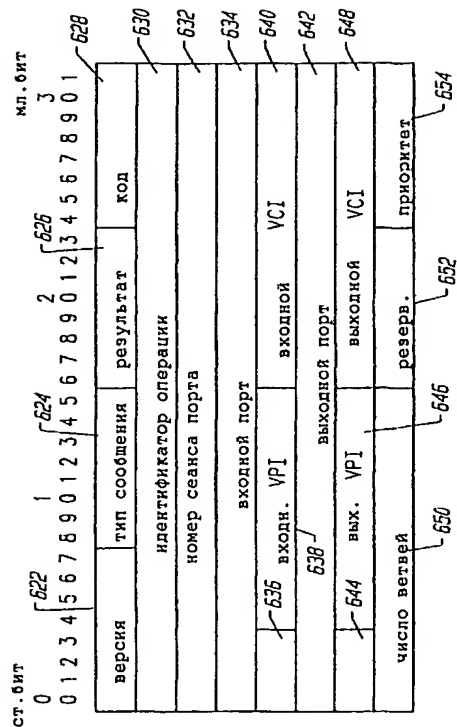
RU 2189072 C2

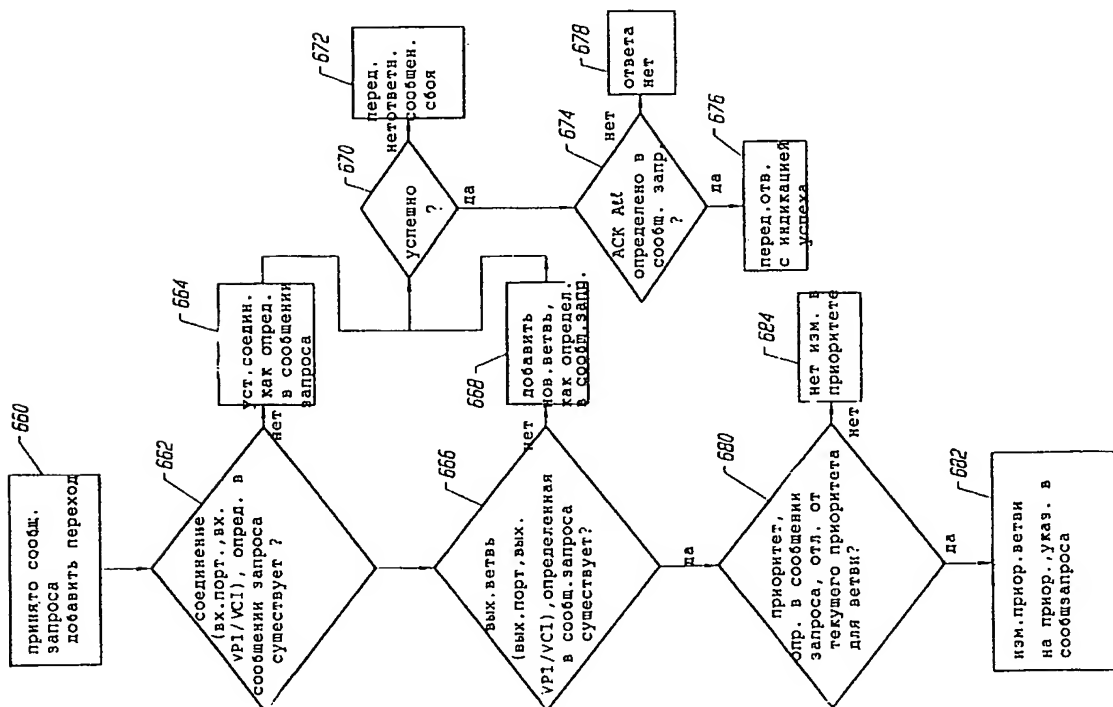
RU 2189072 C2

RU 2189072 C2

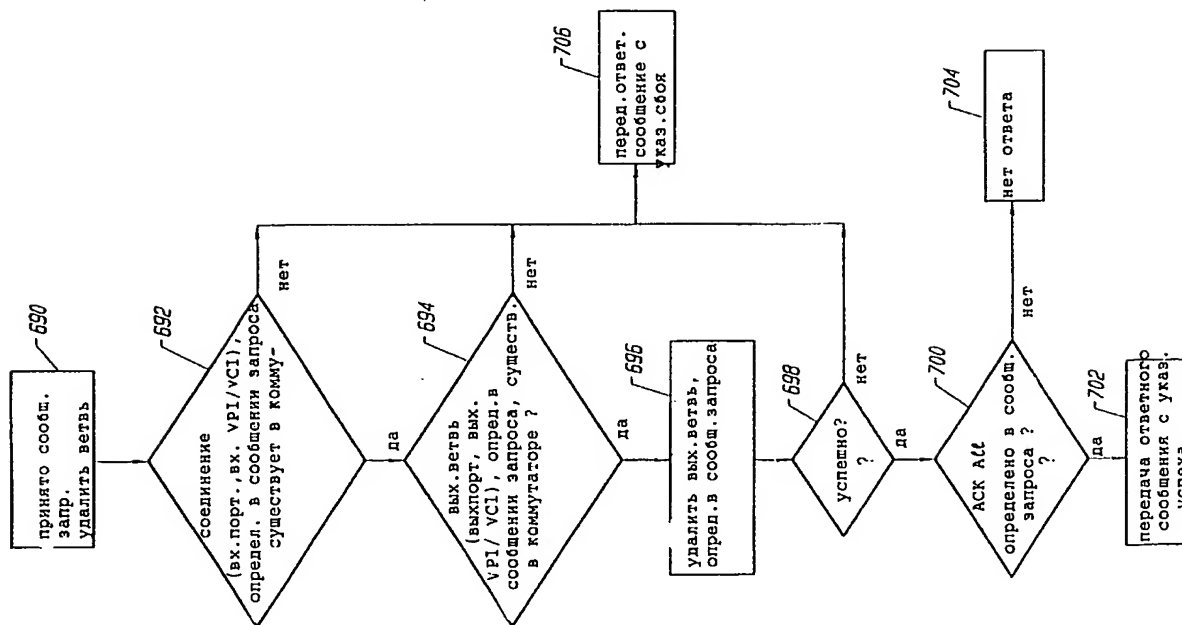


обобщенное сообщение управления
соединением
протокола
620 GSMP

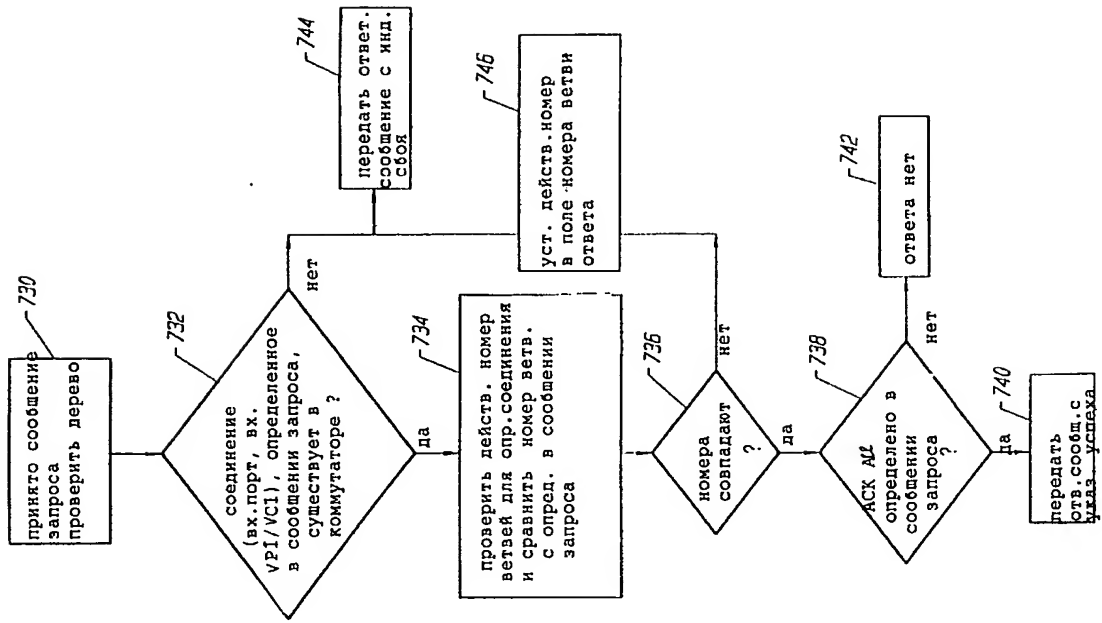




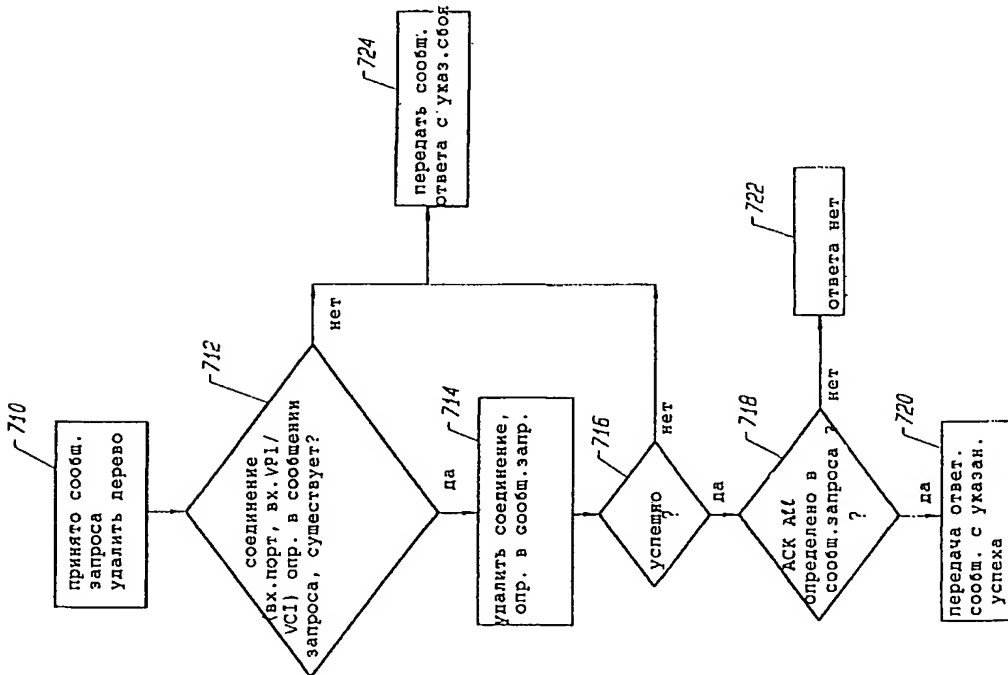
фиг. 13а



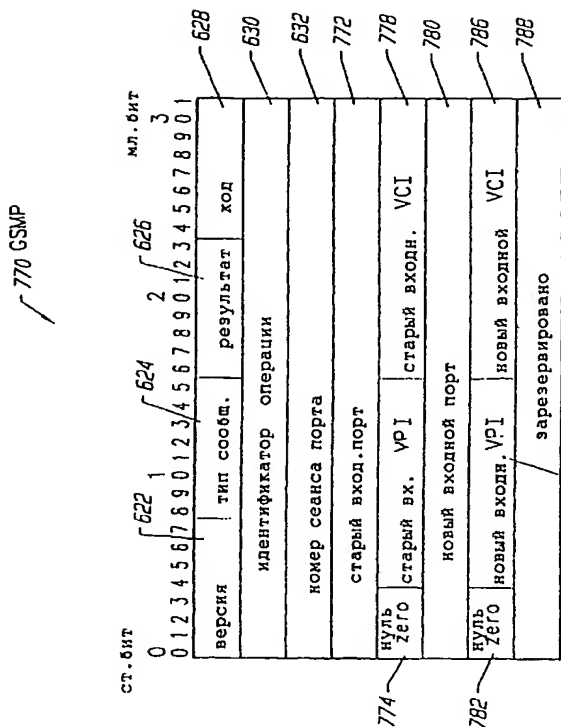
фиг. 13б



фиг. 13d

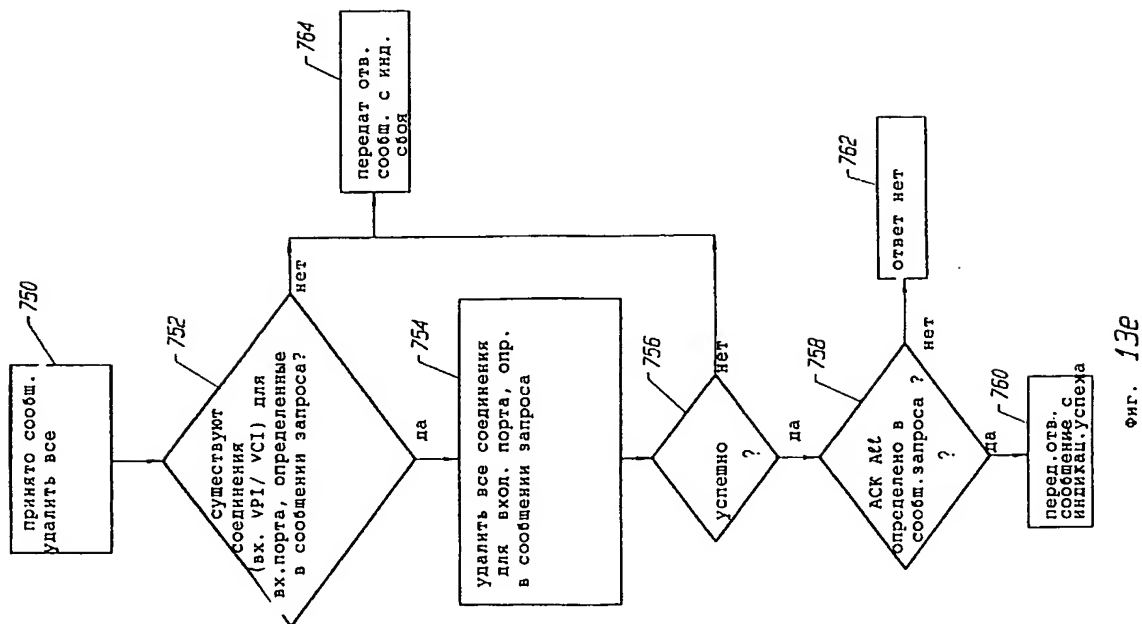


фиг. 13c



RU 2 189 072 C 2

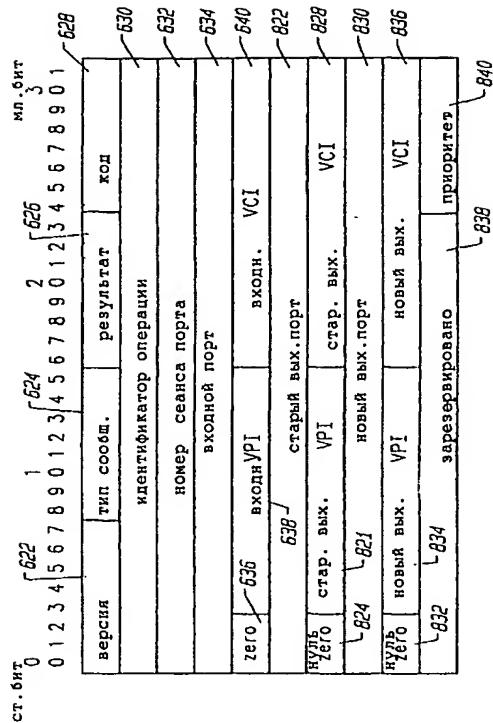
RU 2 189 072 C 2

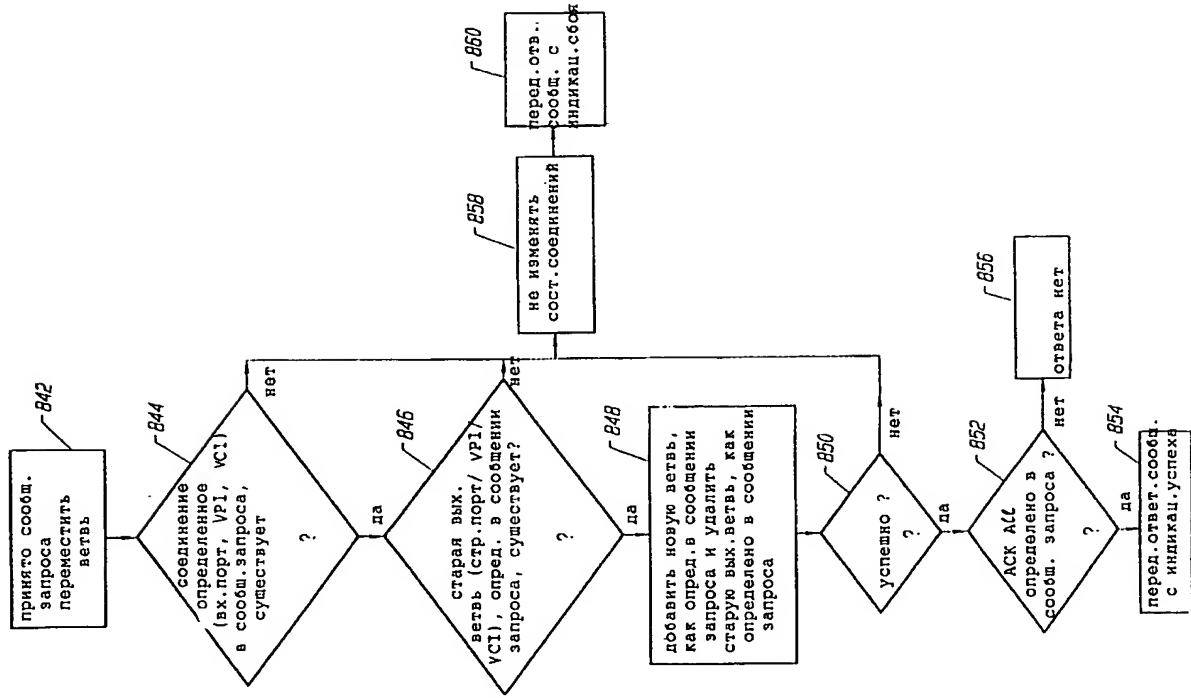
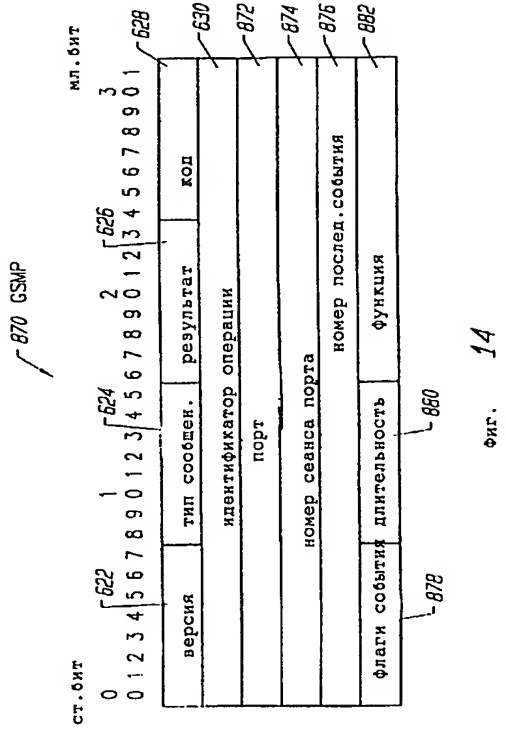


RU 2 189 072 C 2

RU 2 189 072 C 2

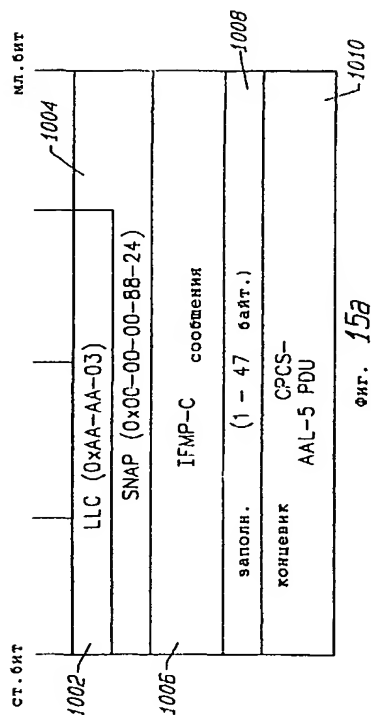
520 CSMP





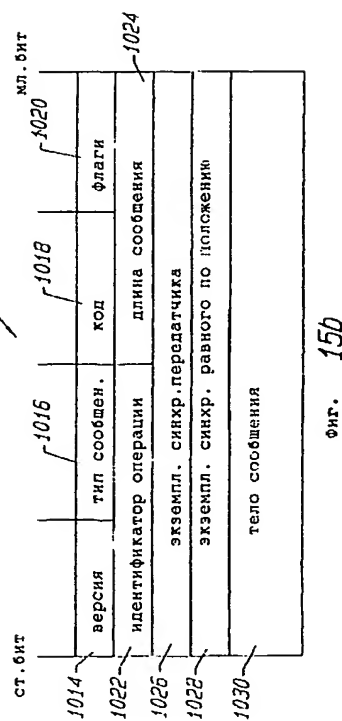
Фиг. 13i

1000 инкапсулированный
IFMP-C пакет

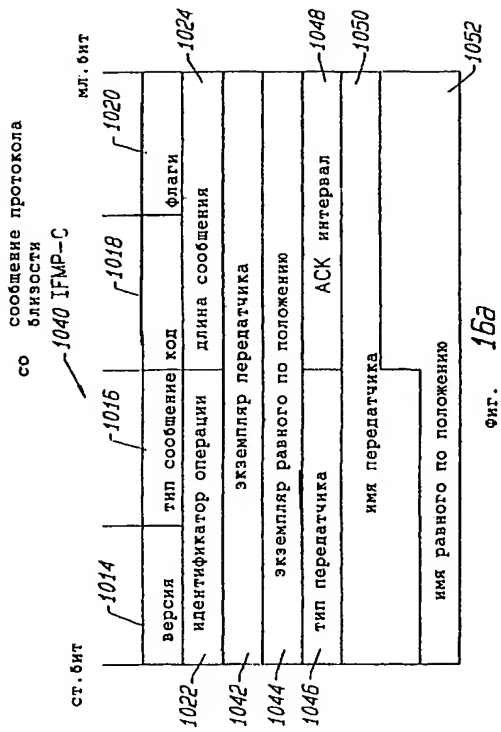


RU 2189072 C2

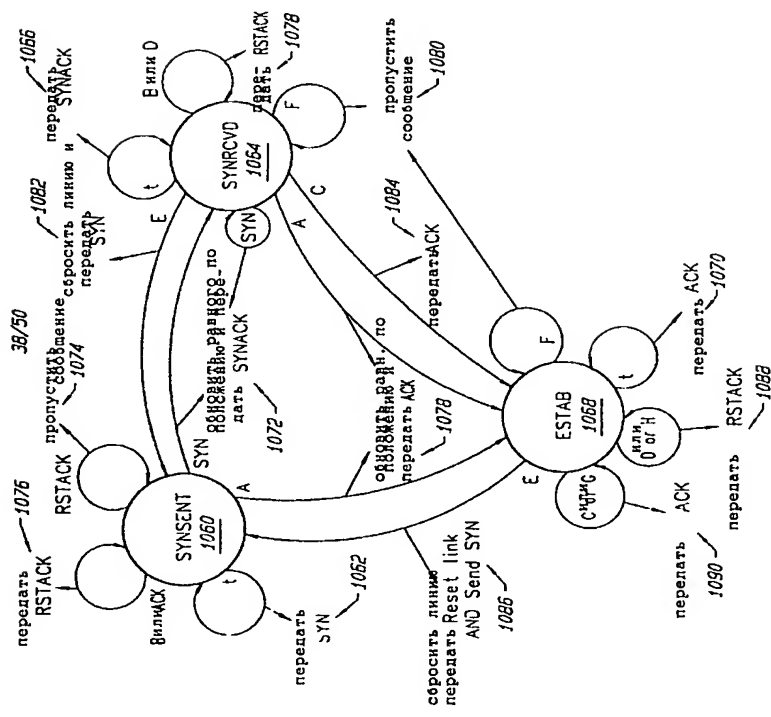
1012 сообщение протокола
IFMP-C



RU 2189072 C2



RU 2189072 C2



- A: SYNACK AND %X
 B: SYNACK AND NOT(%X)
 C: ACK AND %X AND %Y
 D: ACK AND NOT(%X AND %Y)
 E: RSTACK AND %X AND %Y
 F: RSTACK AND NOT(%X AND %Y)
 G: (SYN OR SYNACK) AND NOT(%X)
 H: (SYN OR SYNACK) AND NOT(%X)

Фиг. 16b

1100 IFMP-C

Ст. бит	1014	1022	1026	1028	1112	Мл. бит
1014	версия	тип сообщения	код	флаг		1020
1022	идентиф. операции	экз. синх. передатчика	экз. синх. равного по положению	следующий идентификатор		1024

Фиг. 17a

сообщение ответа перечня
интерфейсов протокола
1114 IFMP-C

1114 IFMP-C

Ст. бит	1014	1022	1026	1028	1112	1116	1118	1120	Мл. бит
1014	версия	тип сообщения	код	флаг					1020
1022	идентиф. операции	экз. синх. передатчика	экз. синх. равного по положению	следующий идентификатор	число интерфейсов	идентиф. 1 интерфейса	идентиф. 2 интерфейса		1024

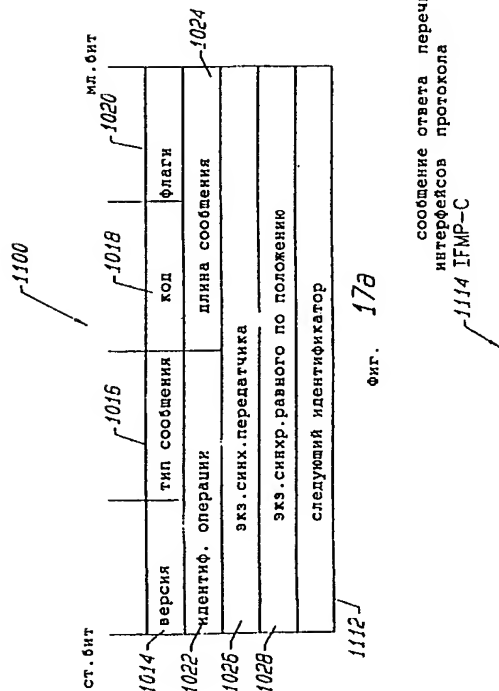
Фиг. 17b

RU 2 1 8 9 0 7 2 C 2

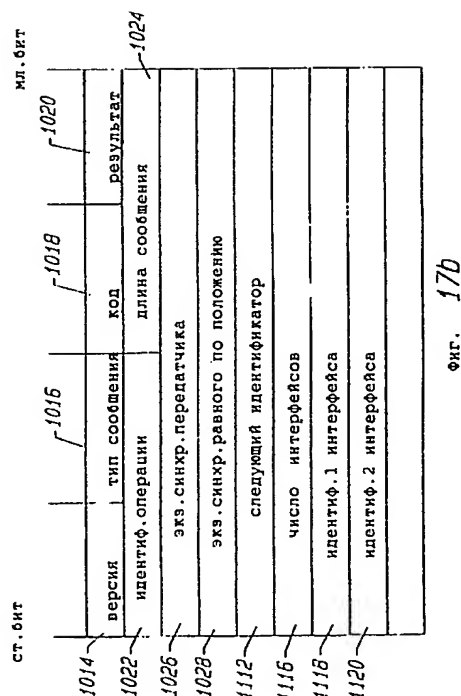
RU 2 1 8 9 0 7 2 C 2

RU 2 1 8 9 0 7 2 C 2

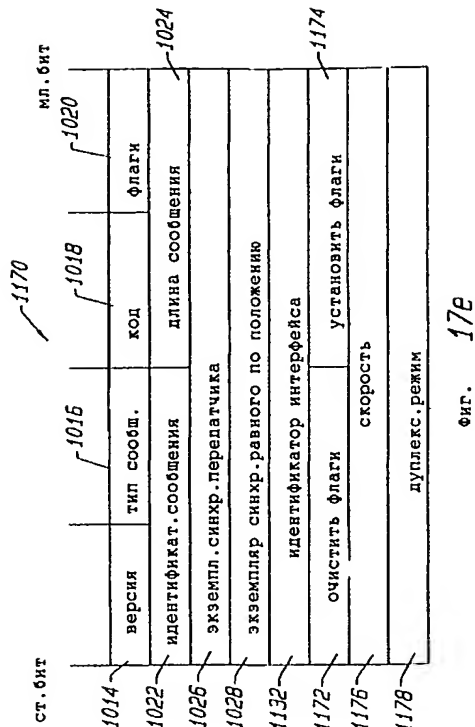
RU 2 1 8 9 0 7 2 C 2



RU 2 189 072 C 2



RU 2 189 072 C 2



RU 2 189 072 C 2

ст. бит		мл. бит	
1014	версия	1016	1018
1022	идентиф. операции	код	1020
1026	идентиф. операции	длина сообщения	1024
1028	экз. синхр. передатчика		
1201	экз. синхр. равного по положению		
1202	идентиф. вх. интерфейса		1204
1262	вх. предпеств.	вх. флаги	
	резерв.		1210
1224	вх. данные ключа		
1226	маска вход. ключа		

фиг. 18d

ст. бит		мл. бит	
1014	версия	1016	1018
1022	идент. операции	код	1020
1026	идент. операции	длина сообщения	1024
1028	экз. синхр. перед		
1252	резерв.	экз. синхр. равного по положению	1254
		вых. счет	

фиг. 18c

ст. бит		мл. бит	
1014	версия	1016	1018
1022	идентиф. операции	код	1020
1026	идентиф. операции	длина сообщения	1024
1028	экз. синхр. передатчика		
1201	экз. синхр. равного по положению		
1202	идентиф. входного интерфейса		1204
1206	вх. предпеств.	вх. флаги	
1208	идентиф. вх. интерфейса		
1212	дл. вх. загол.	резерв.	1210
1214	дл. вх. загол.	длина ключа	1218
1216	дл. вх. загол.	длина преобр.	1222
1224	дл. вх. загол.	дл. преобр.	1226
1226	дл. вх. загол.	дл. преобр.	
1228	дл. вх. загол.	дл. преобр.	
1230	дл. вх. загол.	дл. преобр.	

фиг. 18a

ст. бит		мл. бит	
1242	резерв.	1244	длина усечения

фиг. 18b

1300

ст. бит	1014	1016	1018	мл. бит	1020
	версия	тип сообщ.	код	флаги	
	1022	идент. операции	длина сообщ.		1024
	1026	экз. синхр. перед.			
	1028	экз. синхр. равного по положению			
	1201	идентиф. вх. интерфейса			
	1202	вх. препочт.	вх. флаги		1204
	1302	идентиф. старого вых. интерфейса			
	1304	резерв.	1312	длина ключа	1210
	1308	дл. стар. вх. интерфейса	стар. тип преобр.	дл. стар. вх. интерфейса	1314
	1318	обр. стар. качества обслуж.			
	1320	идентиф. нового вх. интерфейса			
	1322	резерв.	1328		
	1324	дл. нов. вх. интерфейса	нов. тип преобр.	дл. нов. вх. интерфейса	1330
	1334	обр. нов. кач. обслуживания			
	1224	вх. данные ключа			
	1226	маска вх. ключа			
	1340	старый вх. заголовок			
	1342	стар. дан. преобраз.			
	1344	новый вх. заголовок			
	1346	нов. дан. преобраз.			

RU 2189072 C2

фиг. 18e

1400

ст. бит	1014	1016	1018	мл. бит	1020
	версия	тип сообщ.	код	флаги	
	1022	идентиф. операции	длина сообщ.		1024
	1026	экз. синхр. перед.			
	1028	экз. синхр. равного по положению			
	1402	данные 1 дерева			
	1404	данные 2 дерева			

фиг. 19a
структура поля данных дерева

1405

ст. бит	1132	мл. бит
	идентиф. вх. интерфейса	
1202	вх. предшеств.	1204
1408	резерв.	
1210	дл. ключа	размер записи
1412	счет использ.	
1224	вх. данные ключа	
1226	маска вх. ключа	

фиг. 19b

RU 2189072 C2

RU 2189072 C2

1420

ст. бит	1016	1018	мл. бит
1014	версия	тип сообщ.	код
1022	идентиф. операции	длина сообщения	1024
1026	экз. синхр. передатчика		
1028	экз. синхр. равного по положению		
1201	идентиф. вх. интерфейса		
1422	след. идентификатор		

фиг. 20a

RU 2189072 C2

RU 2189072 C2

1430

ст. бит	1016	1018	мл. бит
1014	версия	тип сообщ.	код
1022	идентиф. операции	длина сообщения	1024
1026	экз. синхр. перед.		
1028	экз. синхр. равного по положению		
1201	идент. вх. интерфейса		
1202	вх. предшеств.	вх. флаги	1204
1206	идент. вх. интерфейса		
1432	резерв.		длина ключа
1212	дл. вх. загол.	длина удал.	тип преобр.
1422	1214	обр. качества усл.	1216
1224	след. идентификатор		длина дан. преобр.
1226	вход. данные ключа		1222
1228	маска вх. ключа		
1230	вых. данные загол.		
	данные преобразования		

фиг. 20b

RU 2189072 C2

RU 2189072 C2

1014	версия	тип сообщ.	код	плата	1018	1020	мл. бит
1022	идентиф. операции		длина сообщения		1024		
1026	экз. синхр. передатчика						
1028	экз. синхр. равного по положению						
1472	статистика интерфейсов						
1474	статистика интерфейсов						

Фиг. 21С

структура поля
статистика интерф.

ст. бит	1482	1484	1486	мл. бит
резерв.	тип интерф.	длина записи		
идентифик. интерф.				
1488	длина общей статист.	длина спец. статист.		
1490	1492			
1494	общ. статист.			
1496	спец. статист			

Фиг. 210

1014	версия	тип сообщ.	код	флаги	1020
1022	идентиф. операции				1024
1026	экз. синхр. перед.				
1028	экз. синхр. равного по положению				
1442	идентиф. узла				
1444	исход. идентиф.				
1446	тип узла	резерв.			1448
1450	исх. слот				
1452	исх. стойка				1456
	мин версия	макс. версия			

1454

Фит. 218

Фиг. 21а

-1460

1014	версия	тип сообщ.	код	флаги	1020
1022	идент. операции	длина сообщ.			1024
1026	экз. синхр. перед.				
1028	экз. синхр. равного по положению				
1462	резерв.	число интерф.			1464
1466	идентиф. 1 интерфейса				
1468	идентиф. 2 интерфейса				

Фиг. 21б

1494

ст. бит	структура поля обдстатист.	мл. бит
1500	приним. байты	
1502	принятые пакеты групп.передачи	
1504	принятые пакеты широкополос.перел.	
1506	принят. пакеты однократ.перел.	
1508	принят. отказы	
1510	принятые ошибки	
1512	принят. неизв.	
1514	передав. байты	
1516	перед.пакеты групп.передачи	
1518	перед.пакеты широкополос.передачи	
1520	перед.пакеты однократ.передачи	
1522	перед. отказы	
1524	перед.ошибки	

Фиг. 21e
1530 ATM- структура полей спец.статист
мл. бит

1532	принятые элементы данных
1534	переданные элементы данных
1535	ошибки ЦИК
1536	физич. ошибки

Фиг. 21f структура полей спец.статист
тики для
Ethernet -данных
1540

1542	принятые ошибки ЦИК
1544	конфликты при передаче

Фиг. 21g

RU 2189072 C2

RU 2189072 C2